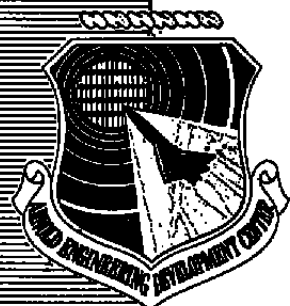


cy.8



A METHOD OF CHARACTERISTICS COMPUTER PROGRAM FOR THREE-DIMENSIONAL SUPERSONIC INTERNAL FLOWS

**W. C. Armstrong
ARO, Inc., a Sverdrup Corporation Company**

**ENGINE TEST FACILITY
ARNOLD ENGINEERING DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
ARNOLD AIR FORCE STATION, TENNESSEE 37389**

January 1979

Final Report for Period October 1977 — September 1978

Approved for public release; distribution unlimited.

Property of U. S. Air Force
AEDC LIBRARY
F40600-77-C-0003

Prepared for

**ARNOLD ENGINEERING DEVELOPMENT CENTER/DOIR
ARNOLD AIR FORCE STATION, TENNESSEE 37389**

NOTICES

When U. S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Qualified users may obtain copies of this report from the Defense Documentation Center.

References to named commercial products in this report are not to be considered in any sense as an indorsement of the product by the United States Air Force or the Government.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

APPROVAL STATEMENT

This report has been reviewed and approved.



ELTON R. THOMPSON
Project Manager, Research Division
Directorate of Test Engineering

Approved for publication:

FOR THE COMMANDER



ROBERT W. CROSSLEY, Lt Colonel, USAF
Acting Director of Test Engineering
Deputy for Operations

UNCLASSIFIED

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AEDC-TR-78-68	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A METHOD OF CHARACTERISTICS COMPUTER PROGRAM FOR THREE-DIMENSIONAL SUPERSONIC INTERNAL FLOWS	5. TYPE OF REPORT & PERIOD COVERED Final Report - October 1977 - September 1978	
	6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s) W. C. Armstrong, ARO, Inc.	8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Arnold Engineering Development Center/DOTR Air Force Systems Command Arnold Air Force Station, Tennessee 37389	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element 65807F	
11. CONTROLLING OFFICE NAME AND ADDRESS Arnold Engineering Development Center/OIS Air Force Systems Command Arnold Air Force Station, Tennessee 37389	12. REPORT DATE January 1979	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. NUMBER OF PAGES 94	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Available in DDC		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) computer programs flow (shock-free) three dimensional nozzles supersonic flow (internal) axially symmetric flow transonic flow		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The reference-plane method of characteristics is applied to the computation of the supersonic flow in a three-dimensional (3-D) channel such as a propulsion nozzle. The fluid is assumed to be an inviscid ideal gas and the flow is assumed to be shock-free, although it can be rotational. Both the basic equations and the numerical procedures are described, as is the computer program, which was written in FORTRAN IV language for the IBM 370/165		

UNCLASSIFIED

UNCLASSIFIED

20. ABSTRACT (Continued)

computer. The validity of the computer program was established by computing, in various ways, an axisymmetric nozzle flow as a three-dimensional flow; the numerical results are in good agreement with the results from a well-established computer program for axisymmetric flow.

PREFACE

The work reported herein was conducted by the Arnold Engineering Development Center (AEDC), Air Force Systems Command (AFSC). The results presented were obtained by ARO, Inc., AEDC Division (a Sverdrup Corporation Company), operating contractor for the AEDC, AFSC, Arnold Air Force Station, Tennessee. Elton R. Thompson was the Air Force project manager. The work was done under ARO Project No. E32A-POA, and the manuscript was submitted for publication on October 5, 1978.

The author of this report is a Computer Applications Analyst in the Computational Services Branch, Facility Support Department, ARO, Inc.

CONTENTS

	<u>Page</u>
1.0 INTRODUCTION	5
2.0 GOVERNING EQUATIONS	
2.1 Equations of Motion	6
2.2 Characteristic Equations	7
2.3 Compatibility Equations	8
3.0 NUMERICAL PROCEDURE	
3.1 Method of Solution	9
3.2 Working Equations	11
4.0 RESULTS AND DISCUSSION	19
5.0 CONCLUDING REMARKS	21
REFERENCES	22

ILLUSTRATIONS

Figure

1. Coordinate System	8
2. Field Point Network	10
3. Body Point Network	18
4. Geometry for Test Cases	19
5. Results for z Axis Aligned with Nozzle Centerline	20
6. Results of 3-D MOC Program	21

APPENDIXES

A. DESCRIPTION OF COMPUTER PROGRAM	23
B. EXAMPLE PROBLEM	70
NOMENCLATURE	93

1.0 INTRODUCTION

Theoretical calculations of exhaust nozzle performance are often required to aid in the evaluation of engines tested at the Arnold Engineering Development Center (AEDC). In the past, most propulsion nozzles have been axisymmetric and well-established computer programs are available for computing the flow field in such nozzles. However, future tests at AEDC will involve engines with three-dimensional (3-D) exhaust nozzles, and no programs have been available for computing the flow field in 3-D nozzles. Consequently, development of 3-D computer programs for both the transonic and supersonic portions of the flow was initiated. This report describes the supersonic computer program, which is based on the method of characteristics (MOC).---Of course, the computer program described herein is not limited to computation of propulsion nozzle performance, but is also applicable to many other 3-D supersonic internal flows.

Except for the boundary-layer region near the wall, supersonic nozzle flow can be computed with adequate accuracy by assuming the fluid to be inviscid and adiabatic. In addition, most nozzle flows do not contain strong shock waves, so the nozzle performance can be adequately predicted by assuming the flow to be shock-free. However, because of flow phenomena upstream of the nozzle entrance, the flow in propulsion nozzles is often significantly rotational. The rotationality of the flow entering the supersonic region persists throughout the flow field.

In this study, the rotational MOC was chosen as the basis of the numerical analysis. The mathematical theory of characteristics is well established, and many computer programs have been based on the method, particularly for planar and axisymmetric flow. Several programs have been developed to compute the flow over 3-D bodies (e.g., Refs. 1 and 2) and special types of 3-D internal flow have been solved with the method (Ref. 3). However, few attempts have been made to compute general 3-D supersonic internal flows with the MOC.

2.0 GOVERNING EQUATIONS

In this section, the governing flow equations and the resulting characteristic equations are presented. These equations are well documented in the literature (e.g., see Ref. 1); therefore, no development will given.

2.1 EQUATIONS OF MOTION

The steady, inviscid, 3-D flow equations for an ideal gas are:

Continuity:

$$u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + w \frac{\partial \rho}{\partial z} + \rho \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = 0 \quad (1)$$

Momentum:

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial x} = 0 \quad (2)$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial y} = 0 \quad (3)$$

$$u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial z} = 0 \quad (4)$$

Energy:

$$u \frac{\partial h}{\partial x} + v \frac{\partial h}{\partial y} + w \frac{\partial h}{\partial z} - \frac{1}{\rho} \left(u \frac{\partial p}{\partial x} + v \frac{\partial p}{\partial y} + w \frac{\partial p}{\partial z} \right) = 0 \quad (5)$$

State:

$$p = \rho RT \quad (6)$$

2.2 CHARACTERISTIC EQUATIONS

The equations of motion give rise to two sets of characteristic surfaces. The defining equations are

$$(uf_x + vf_y + wf_z)^2 = 0 \quad (7)$$

and

$$(ug_x + vg_y + wg_z)^2 - a^2 (g_x^2 + g_y^2 + g_z^2) = 0 \quad (8)$$

where $f(x, y, z) = 0$ and $g(x, y, z) = 0$ are the characteristic surfaces. The first surface is composed of streamlines and the second is the Mach conoid. The equation of a ray, or bicharacteristic, of the Mach conoid may be expressed as

$$dx = (\cos \beta \sin \theta - \sin \beta \cos \theta \cos \delta) dL \quad (9)$$

$$dy = (\cos \beta \cos \theta \sin \psi - \sin \beta (\sin \theta \sin \psi \cos \delta - \cos \psi \sin \delta)) dL \quad (10)$$

$$dz = (\cos \beta \cos \theta \cos \psi - \sin \beta (\sin \theta \cos \psi \cos \delta + \sin \psi \sin \delta)) dL \quad (11)$$

where β is the Mach angle, dL is the distance along the bicharacteristic, δ is a parametric angle, and θ and ψ are related to the velocity vector by

$$u = q \sin \theta \quad (12)$$

$$v = q \cos \theta \sin \psi \quad (13)$$

$$w = q \cos \theta \cos \psi \quad (14)$$

The parametric angle δ lies in a plane normal to the velocity vector and is measured from the plane containing \bar{q} and x . The relationships between the variables are shown in Fig. 1.

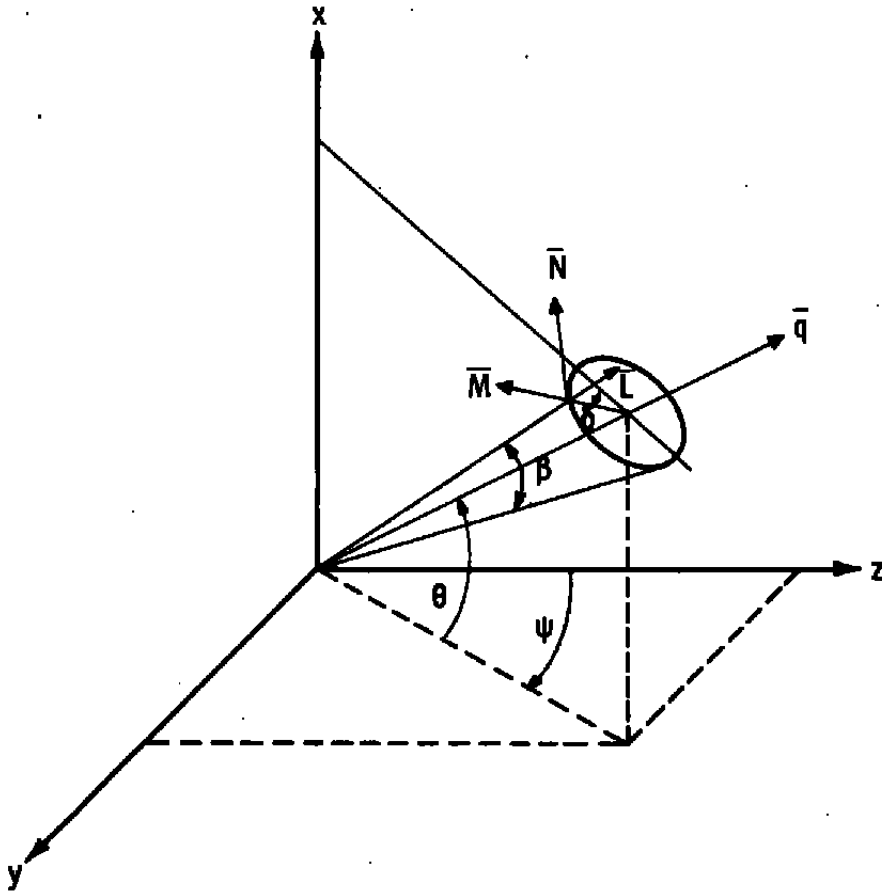


Figure 1. Coordinate system.

2.3 COMPATIBILITY EQUATIONS

The compatibility equations are determined by the flow equations and the requirement that the derivatives in a direction normal to the characteristic surface disappear. The compatibility equation which applies along the Mach conoid is

$$\frac{\cot \beta}{\rho q^2} \frac{\partial p}{\partial L} + \cos \delta \frac{\partial \theta}{\partial L} + \cos \theta \sin \delta \frac{\partial \psi}{\partial L} - \sin \rho \left(\cos \theta \cos \delta \frac{\partial \psi}{\partial N} - \sin \delta \frac{\partial \theta}{\partial N} \right) = 0 \quad (15)$$

where $\frac{\partial}{\partial L}$ and $\frac{\partial}{\partial N}$ are derivatives along and normal to the bicharacteristic.

The compatibility equations along a streamline are

$$\frac{\gamma}{\gamma - 1} R dT = \frac{1}{\rho} dp \quad (16)$$

and

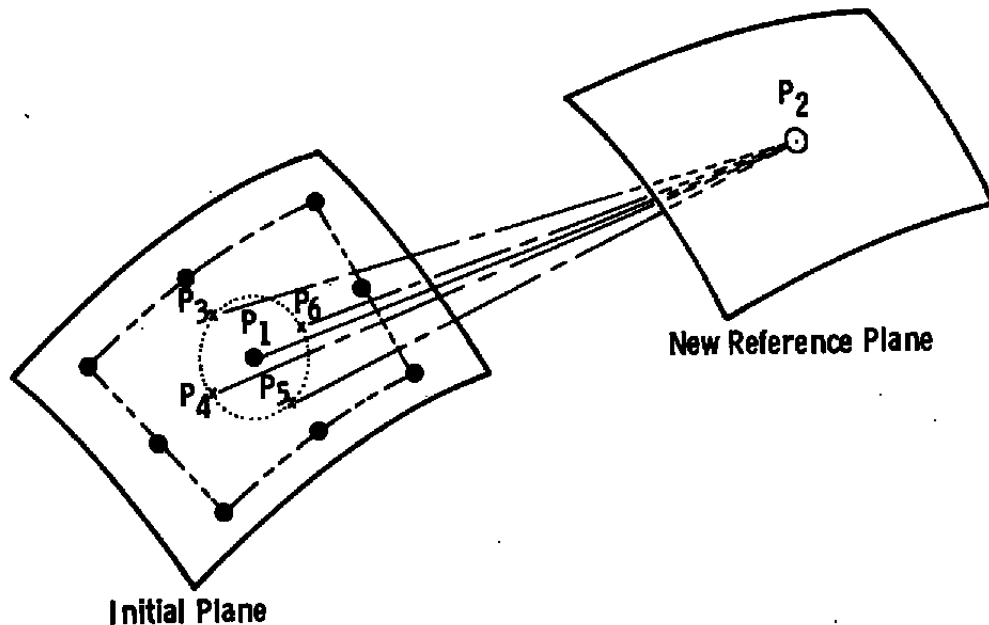
$$\frac{1}{\rho} dp = -q dq \quad (17)$$

3.0 NUMERICAL PROCEDURE

The numerical procedure is similar to the one developed by Strom (Ref. 1) for external flow. The method traces streamlines from a known reference plane to the next reference plane as illustrated in Fig. 2 for a field point. The flow properties at approximately equally spaced points on an initial plane normal to the z axis are assumed known. The reference planes throughout the flow field are assumed to be normal to the z axis.

3.1 METHOD OF SOLUTION

A new reference plane is located a distance dz from the initial plane. This distance must be determined such that the Courant-Friedrichs-Lewy (C-F-L) stability conditions are satisfied (Refs. 1 and 4). The C-F-L stability conditions are satisfied if dz is smaller than the minimum intersection distance of the Mach conoids from the initial points.



- Known Points in Initial Plane
- Point in New Reference Plane (P_2)
- x Points in Initial Plane on Bicharacteristics from P_2
- Streamline
- Bicharacteristic
- Domain of Dependence of Difference Network
- Domain of Dependence of Differential Equations

For the C-F-L stability conditions to be satisfied, the domain of dependence of the differential equations must be contained within the domain of dependence of the difference network.

Figure 2. Field point network.

The intersection of a streamline from a point P_1 in the initial plane with the new reference plane locates point P_2 . Four bicharacteristics, 90 deg apart, from P_2 to the initial plane yield points P_3 , P_4 , P_5 , and P_6 .

The properties at P_1 and eight surrounding neighbors in the initial plane are surface fitted. The flow conditions at points $P_3 - P_6$ are evaluated from the surface fit. The compatibility equations are applied

along the bicharacteristics and the streamline to obtain the flow conditions at P_2 . This procedure is iterated until the flow conditions at P_2 converge. When all points on the new plane are determined, the new plane becomes the initial plane for the next calculation.

3.2 WORKING EQUATIONS

Several of the computational steps are the same for both field points and body points. These common phases of the procedure will be discussed first. Then the computational steps required for field and body points will be presented individually.

3.2.1 Common Procedures

The surface fit of the flow properties at P_1 and eight neighboring points in the initial plane is made once in the iteration procedure to obtain P_2 and its properites. The fit chosen is a spline surface fit (Ref. 5) which fits a variable w as a function of x and y :

$$w(x,y) = a_0 + a_1 x + a_2 y + \sum_{i=1}^9 b_i r_i^2 \ln r_i^2 \quad (18)$$

where

$$r_i^2 = (x - x_i)^2 + (y - y_i)^2$$

Five variables are fit: p , ρ , q , θ , and ψ . The surface fit is used to evaluate the variables at $P_3 - P_6$. The partial derivatives $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ are also evaluated using this fit:

$$\frac{\partial w}{\partial x} = a_1 + 2 \sum_{i=1}^9 b_i (1 + \ln r_i^2) (x - x_i) \quad (19)$$

$$\frac{\partial w}{\partial y} = a_2 + 2 \sum_{i=1}^9 b_i (1 + \ln r_i^2)(y - y_i) \quad (20)$$

The derivatives $\frac{\partial \theta}{\partial N}$ and $\frac{\partial \psi}{\partial N}$ used in the compatibility equation along a bicharacteristic must be obtained. The procedure for evaluating them is the same; therefore, only the procedure to obtain $\frac{\partial \theta}{\partial N}$ will be given. Write $\frac{\partial \theta}{\partial N}$ as

$$\left(\frac{\partial \theta}{\partial N}\right)_i = \left(\frac{\partial \theta}{\partial x}\right)_i \left(\frac{\partial x}{\partial N}\right)_i + \left(\frac{\partial \theta}{\partial y}\right)_i \left(\frac{\partial y}{\partial N}\right)_i + \left(\frac{\partial \theta}{\partial z}\right)_i \left(\frac{\partial z}{\partial N}\right)_i \quad (21)$$

The derivatives $\frac{\partial \theta}{\partial x}$ and $\frac{\partial \theta}{\partial y}$ are obtained from the surface fit. Then, $\frac{\partial \theta}{\partial z}$ is obtained by writing the equation as a difference equation and solving for $\frac{\partial \theta}{\partial z}$:

$$\theta_2 - \theta_1 = \left(\frac{\partial \theta}{\partial x}\right)_1 (x_2 - x_1) + \left(\frac{\partial \theta}{\partial y}\right)_1 (y_2 - y_1) + \left(\frac{\partial \theta}{\partial z}\right)_1 dz \quad (22)$$

The derivatives normal to the bicharacteristic are obtained from the coordinate transformation as

$$\left(\frac{\partial x}{\partial N}\right)_i = -\cos \theta \sin \delta_i \quad (23)$$

$$\left(\frac{\partial y}{\partial N}\right)_i = \sin \theta \sin \psi \sin \delta_i + \cos \psi \cos \delta_i \quad (24)$$

$$\left(\frac{\partial z}{\partial N}\right)_i = \sin \theta \cos \psi \sin \delta_i - \sin \psi \cos \delta_i \quad (25)$$

The final form of $\frac{\partial \theta}{\partial N}$ becomes

$$\begin{aligned} \left(\frac{\partial \theta}{\partial N}\right)_i &= \left(\frac{\partial \theta}{\partial x}\right)_i \left(\frac{\partial x}{\partial N}\right)_i + \left(\frac{\partial \theta}{\partial y}\right)_i \left(\frac{\partial y}{\partial N}\right)_i \\ &- \left[\theta_1 + \left(\frac{\partial \theta}{\partial x}\right)_1 (x_2 - x_1) + \left(\frac{\partial \theta}{\partial y}\right)_1 (y_2 - y_1) \right] \left(\frac{\partial z}{\partial N}\right)_i / dz + \theta_2 \left(\frac{\partial z}{\partial N}\right)_i / dz \end{aligned} \quad (26)$$

For the first iteration, the flow properties at P_2 are assumed to be the same as at P_1 . Thereafter, the last computed values are used. In the sections on the field point and body point, the average values of flow properties are the averages of the two points involved in the calculation (for example, $\theta = (\theta_1 + \theta_2)/2$).

3.2.2 Field Point Routine

Given an initial field point P_1 and eight neighboring points in the initial plane (Fig. 2), spline surface fits of p , ρ , q , θ , and ψ as functions of x and y are made. The iteration for the point P_2 and its flow properties consists of the following steps.

Step 1.

The intersection of the streamline from P_1 with the new reference plane locates the new field point P_2 .

$$x_2 = x_1 - \sin \theta \, dz / (\cos \theta \cos \psi) \quad (27)$$

$$y_2 = y_1 + \sin \psi \, dz / \cos \psi \quad (28)$$

$$z_2 = z_1 + dz \quad (29)$$

Initially θ and ψ are the values at P_1 . Successive iterations use the average of θ and ψ at P_1 and P_2 .

Step 2

Four bicharacteristics extended from P_2 to the initial plane give the base points P_3 , P_4 , P_5 , and P_6 . The flow properties at P_2 are

assumed to be the same as those at P_1 for the first iteration. Thereafter, the flow properties at P_2 are the last values computed. The bicharacteristics are located at the parametric angles $\delta = 0, \frac{\pi}{2}, \pi$, and $\frac{3\pi}{2}$. The equations for the base points are

$$x_i = x_2 - (\cos \beta \sin \theta + \sin \beta \cos \theta \cos \delta_i) dL_i \quad (30)$$

$$y_i = y_2 - [\cos \beta \cos \theta \sin \psi - \sin \beta (\sin \theta \sin \psi \cos \delta_i - \cos \psi \sin \delta_i)] dL_i \quad (31)$$

$$z_i = z_1 \quad (32)$$

$$dL_i = dz / [\cos \beta \cos \theta \cos \psi - \sin \beta (\sin \theta \cos \psi \cos \delta_i + \sin \psi \sin \delta_i)] \quad (33)$$

For the first iteration, the angles β , θ , ψ , and δ are those at P_2 . Successive iterations use the average of the values at P_2 and P_1 .

Step 3

The compatibility equations along the bicharacteristics are solved for p , θ , and ψ at P_2 . Only three of the bicharacteristics are required; however, to improve accuracy, four solutions are obtained using three bicharacteristics at a time. The results of the four solutions are averaged to obtain the values of p , θ , and ψ . The compatibility equation in difference form is

$$\begin{aligned} \frac{\cot \beta_i}{\rho_i q_i^2} (P_2 - P_i) + \cos \delta_i (\theta_2 - \theta_i) + \cos \theta_i \sin \delta_i (\psi_2 - \psi_i) \\ + \sin \beta_i \left(\cos \theta_i \cos \delta_i \left(\frac{\partial \psi}{\partial N} \right)_i - \sin \delta_i \left(\frac{\partial \theta}{\partial N} \right)_i \right) dL_i = 0 \end{aligned} \quad (34)$$

The equation contains the three unknowns P_2 , θ_2 , and ψ_2 . The compatibility equations using P_3 , P_4 , and P_5 are solved simultaneously for P_{21} , θ_{21} , and ψ_{21} - similarly, (P_4, P_5, P_6) , (P_5, P_6, P_3) , and (P_6, P_3, P_4) are used to obtain values of p , θ , and ψ . Thus, $P_2 = (P_{21} + P_{22} + P_{23} + P_{24})/4$. In the compatibility equation, β_i , ρ_i , q_i , and δ_i are the average of the values at P_2 and P_i .

The variables ρ_2 , T_2 , and q_2 are obtained from the compatibility equations along the streamline. The flow along a streamline without shocks is isentropic. Therefore, the equations integrate to the following forms for an ideal gas:

$$T_2 = T_{ts} \left(\frac{p_2}{p_{ts}} \right)^{\frac{\gamma-1}{\gamma}} \quad (35)$$

$$\rho_2 = R T_2 / P_2 \quad (36)$$

$$q_2 = \sqrt{2 \frac{\gamma}{\gamma-1} (T_{ts} - T_2)} \quad (37)$$

where T_{ts} and p_{ts} are the stagnation values for the streamline.

Step 4

The preceding steps are repeated until the flow conditions at P_2 converge. If there is no convergence after 25 iterations, the results are checked with a reduced convergence criterion. If this is satisfied, the point is accepted and a message printed. Otherwise, the program terminates.

3.2.3 Body Point Routine

Given the properties of an initial body point P_1 and those of eight neighboring points on the initial plane (Fig. 3), spline surface fits are made. The body surface is assumed to be given by

$$B(x, y, z) = 0 \quad (38)$$

where B is either a known function or a surface fitting element in the region of the body point. The iteration for the new body point P_2 , in the new reference plane, and its flow properties consists of the following steps:

Step 1

The new body point P_2 is located at the intersection of a plane through P_1 , defined by the body unit normal and the unit velocity vector tangent to the body at P_1 , with the body surface at the new reference plane. This requires the simultaneous solution of the following equations:

$$B(x, y, z) = 0 \quad (39)$$

and

$$\begin{aligned} (n_3 \cos \theta \cos \psi - n_1 \cos \theta \sin \psi)(x - x_1) + (n_1 \sin \theta \\ - n_2 \cos \theta \cos \psi)(y - y_1) = (n_3 \sin \theta - n_1 \cos \theta \sin \psi) dz \end{aligned} \quad (40)$$

where n_1 , n_2 , and n_3 are unit normals to the body surface. The first iteration uses the values of θ , ψ , n_1 , n_2 , and n_3 , at P_1 . Successive iterations use the averages of the properties at P_1 and P_2 .

Step 2

This step is similar to Step 2 of the field point routine. However, only three bicharacteristics are used for the body point calculations. The parametric angles used are

$$\delta_3 = \arccos(-n_1 \sin \theta \cos \psi + n_2 \cos \theta - n_3 \sin \theta \sin \psi) \quad (41)$$

$$\delta_4 = \delta_3 - \pi/2 \quad (42)$$

$$\delta_5 = \delta_3 - \pi/2 \quad (43)$$

Step 3

The compatibility equations along two bicharacteristics are solved simultaneously with the condition for flow tangency to the surface at P_2 . The equation for flow tangency, which provides an additional relationship between θ_2 and ψ_2 may be expressed as

$$n_{12} \cos \theta_2 \cos \psi_2 + n_{22} \sin \theta_2 + n_{32} \cos \theta_2 \sin \psi_2 = 0 \quad (44)$$

where n_{12} , n_{22} , and n_{32} are unit normals to the body at P_2 . A Newton-Raphson routine is used to solve this system of equations. The base points P_3 and P_4 are used to obtain p_{21} , θ_{21} , and ψ_{21} ; P_3 and P_5 are used to obtain p_{22} , θ_{22} , and ψ_{22} . These two results are averaged to obtain the values at p_2 ($p_2 = (p_{21} + p_{22})/2$).

The compatibility equations along the streamline are solved the same way as for the body point yielding ρ_2 , q_2 , and T_2 .

Step 4

The iteration procedure for the body point is the same as the procedure used for the field point.

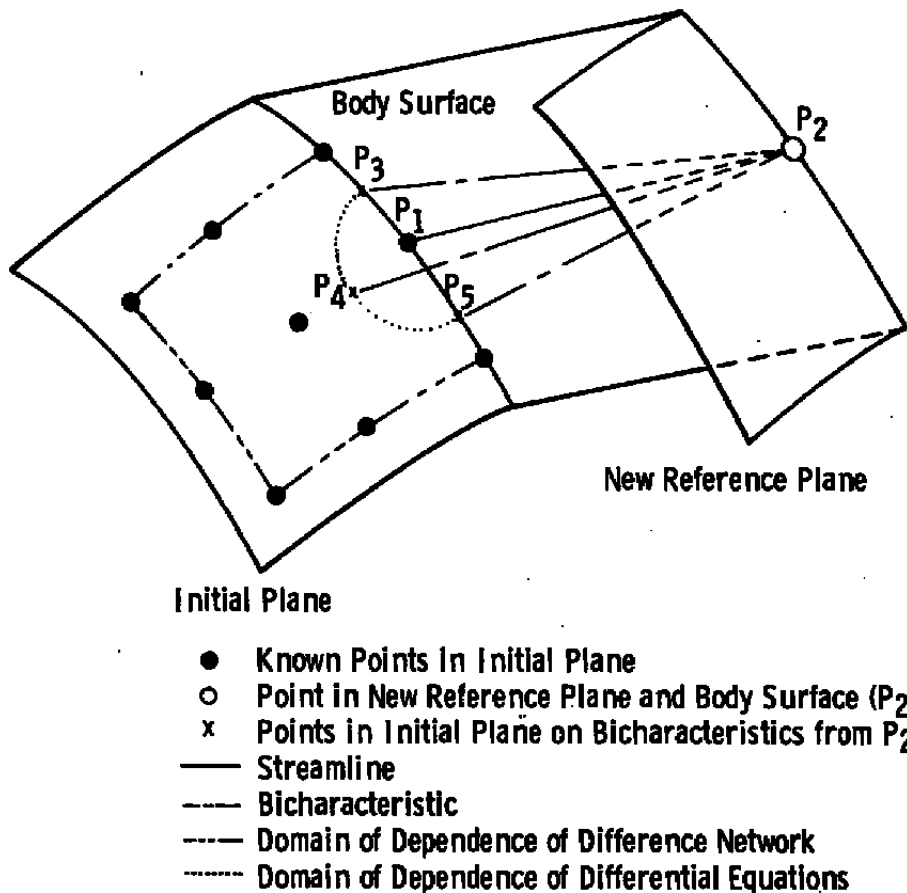


Figure 3. Body point network.

3.2.4 Computer Program

The preceding equations were programmed in FORTRAN IV language for solution on an IBM 370/165 computer. A description and listing of the program is given in Appendix A, and an example problem is given in Appendix B. The program requires a storage capacity of approximately 192,000 bytes.

4.0 RESULTS AND DISCUSSION

The present 3-D MOC program was evaluated by computing the flow in a typical axisymmetric nozzle. With $\gamma = 1.24$ and uniform flow in the throat, the nozzle produces an exit Mach number of 4.1. The 3-D MOC computations were done in three different ways (see 4).

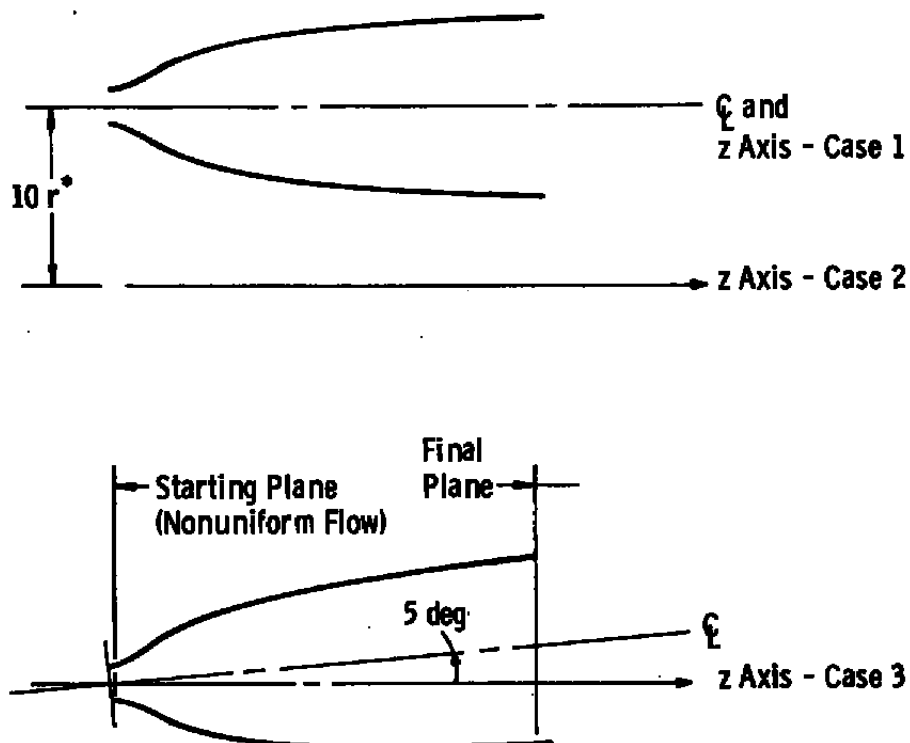


Figure 4. Geometry for test cases.

Case 1 - The z axis was aligned with the geometric centerline of the nozzle.

Case 2 - The z axis was parallel to the nozzle centerline but was displaced by a distance of 10 nozzle throat radii; thus, the z axis for the flow field is completely outside the nozzle.

Case 3 - The z axis was rotated 5 deg relative to the nozzle centerline. In this case, the flow in the starting plane is nonuniform and was obtained by interpolation from the flow field predicted for case 1.

The predicted wall Mach number distribution for case 1 is shown in Fig. 5 along with the predictions of the Lockheed axisymmetric MOC computer program (Ref. 6) which is well verified and widely used. The results from the two programs are in good agreement. At the nozzle exit plane, the circumferential variation of wall Mach number calculated by the 3-D MOC program is much less than one percent.

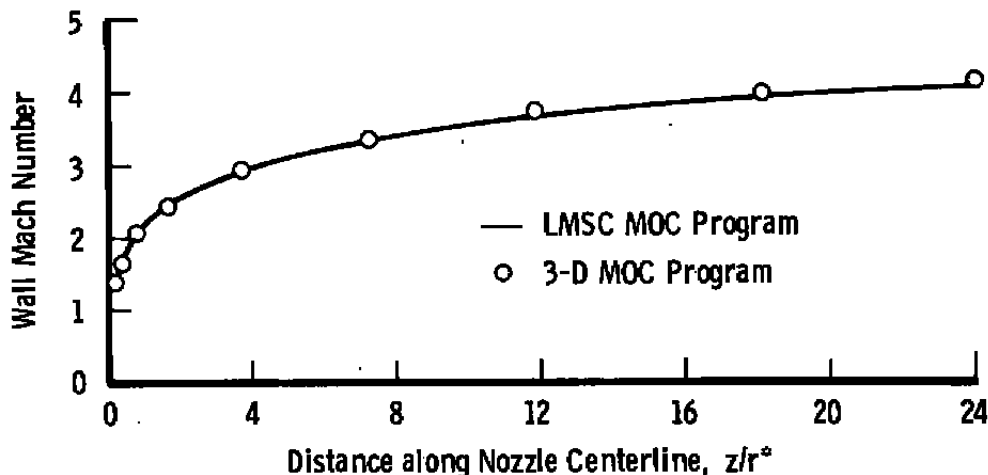


Figure 5. Results for z axis aligned with nozzle centerline.

The results of the three 3-D computations are compared in Fig. 6 where it is seen that the three predicted wall Mach number distributions are essentially identical.

These computations of axisymmetric nozzle flow, done the "hard way" with the 3-D MOC program, indicate that the program is acceptably accurate and is capable of computing the supersonic flow in complex configurations.

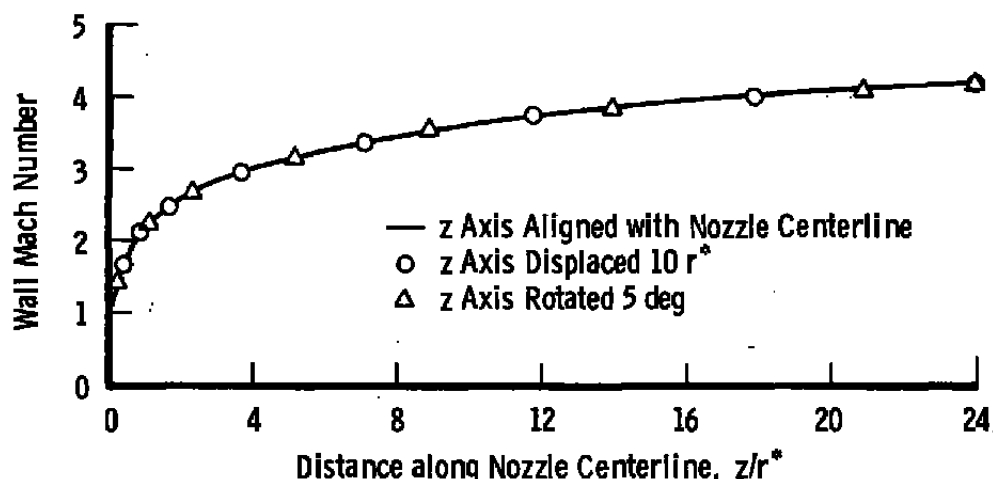


Figure 6. Results of 3-D MOC program.

In each of the test cases, 346 streamlines were traced throughout the flow field. For cases 1 and 2, 78 reference planes were computed which required a CPU time of approximately 25 min. Fewer reference planes were computed for case 3, with a corresponding reduction in CPU time.

5.0 CONCLUDING REMARKS

The results obtained for an axisymmetric nozzle with the 3-D MOC computer program are acceptably accurate. However, additional verification of the program is clearly required. The predictions of the code should be compared with experimental results for 3-D supersonic flows, whenever adequately detailed experimental results become available.

Two extensions of the 3-D MOC computer program are recommended. First, the program should be modified to include a constant-pressure free boundary condition for part of the flow field. This modification would permit the computation of the exhaust plume from a 3-D nozzle. The second recommended modification is the inclusion of shock formation, both within the nozzle and in the exhaust plume.

REFERENCES

1. Strom, Charles R. "The Method of Characteristics for Three-Dimensional Real Gas Flows." AFFDL-TR-67-47 (AD661342), July 1967.
2. Tsung, Cheng Chin. "Study of Three-Dimensional Supersonic Flow Problems by a Numerical Method Based on the Method of Characteristics." Ph.D. Thesis, University of Illinois, 1960.
3. Armstrong, W. C. and Bauer, R. C. "Analysis of Three-Dimensional Inviscid Supersonic Flow Between a Body and an Outer Wall (With Application to a Jet Stretcher System)." AEDC-TR-76-103 (ADA029123), August 1976.
4. Courant, R. and Hilbert, D. Methods of Mathematical Physics, Volume II, Partial Differential Equations. Interscience Publishers, New York and London, 1962.
5. Harder, Robert L. and Demarais, Robert N. "Interpolation Using Surface Splines." Journal of Aircraft, Volume 9, No. 2, February 1972, pp. 189-191.
6. Prozan, R. J. "Development of a Method of Characteristics Solution for Supersonic Flow of an Ideal, Frozen, or Equilibrium Reacting Gas Mixture." Technical Report LMSC/HREC A782535, Lockheed Missiles and Space Co., April 1966.

APPENDIX A DESCRIPTION OF COMPUTER PROGRAM

DESCRIPTION OF ROUTINES

NAME	FUNCTION
MAIN	CALLS CARDIN, INGEOM, INPUT, AND THREED
BDYFIT	CURVE FITS BODY IN AREA OF INTEREST
BODY	BODY POINT ROUTINE
CARDIN	READS AND PRINTS INPUT CARD IMAGES
CUT	INTERPOLATES THE BODY GEOMETRY IN CURRENT PLANE
DELTAF	COMPUTES THE BICHARACTERISTIC ANGLE DELTA
DIST	COMPUTES MIN. DIST. OF CHARACTERISTIC INTERSECTIONS
FIELD	FIELD POINT ROUTINE
FIND	LOCATES POINTS NEAREST TO GIVEN POINT
FINDBP	LOCATES BODY GEOMETRY POINT NEAREST GIVEN POINT
FIT	SURFACE FIT ROUTINE
IDENT	WRITES HEADER AND TRAILER LABELS ON PLOTS
INGEOM	READS GEOMETRY INPUT
INPUT	READS STARTING PLANE INPUT
NEIGH	LOCATES (OR READS) 8 NEIGHBORS OF EACH INPUT POINT
NEWRAP	NEWTON-RAPHSON ROUTINE
NORMAL	COMPUTES THE NORMAL VECTOR TO THE BODY SURFACE
OUTPUT	CONTROLS PRINTED, PLOTTED, AND TAPE OUTPUT
SIMQ	SOLVES SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS
SOLVBP	SOLVES FOR THE LOCATION OF A BODY POINT
SORT	SORT ROUTINE
THREED	CONTROLS THE 3-D CALCULATIONS

INPUT NOMENCLATURE

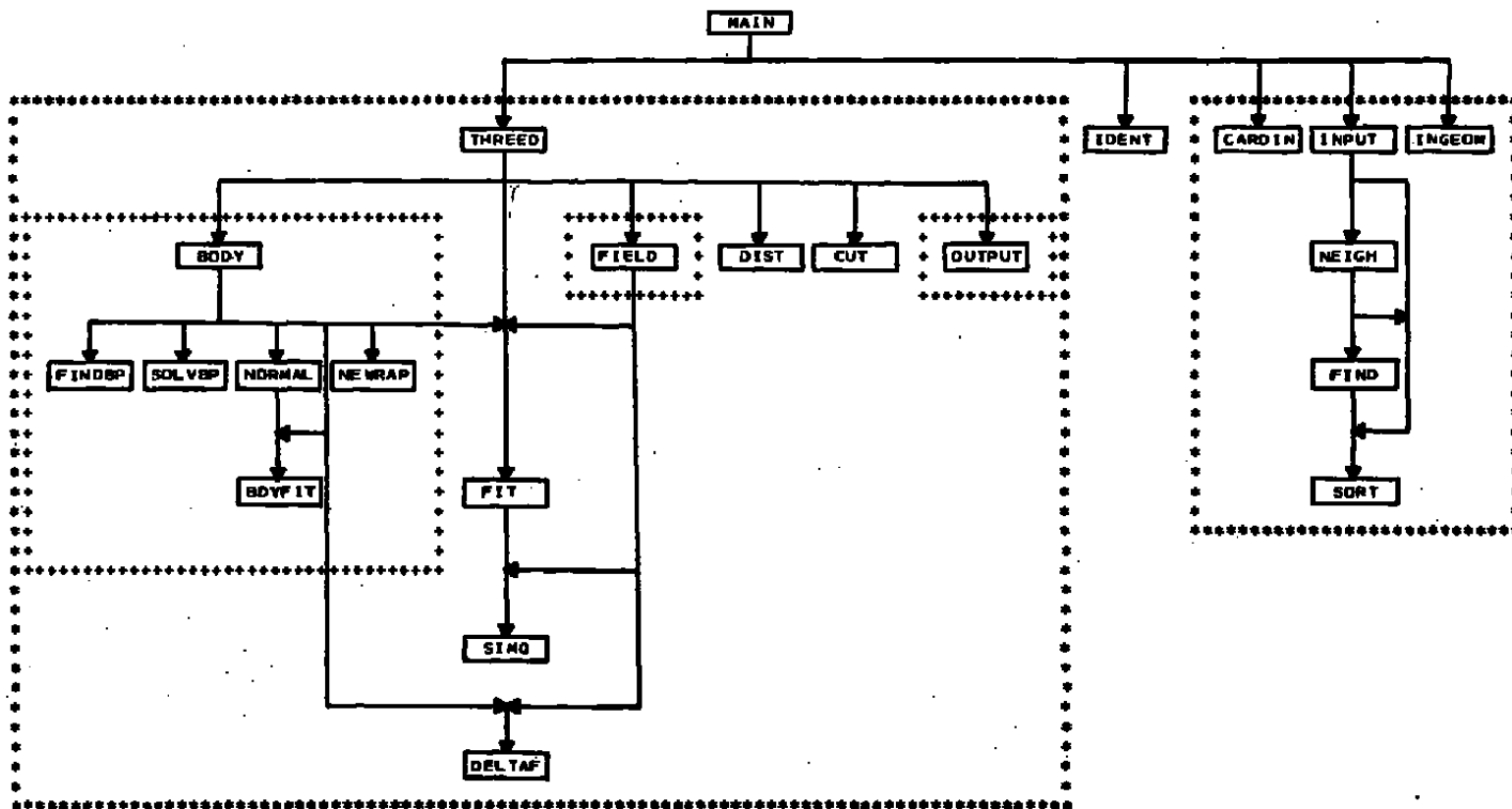
G RATIO OF SPECIFIC HEATS
HEAD PROBLEM DESCRIPTION
IDBUG(I) SPECIAL PRINT FLAGS =0 NO PRINT; =1 PRINT
 IDBUG(1) PRINT INPUT GEOMETRY
 IDBUG(2) PRINT FINAL GEOMETRY
 IDBUG(3) PRINT INITIAL PLANE INPUT
 IDBUG(4) PRINT FINAL INITIAL PLANE DATA
 IDBUG(5) PRINT NEIGHBORS
 IDBUG(11) PRINT NORMAL CALCULATION
IMV INITIAL PLANE DATA TYPE
 IMV=0 Q THETA PSI
 IMV=1 M THETA PSI
 IMV=2 W U V
INEIGH NEIGHBOR CONTROL
 INEIGH=0 NEAREST POINTS ARE NEIGHBORS
 INEIGH=1 SPECIAL COMPUTATION OF NEIGHBORS
 INEIGH=2 READ NEIGHBORS FROM CARDS
 IF ITYPE=0 AND INEIGH=0, INEIGH IS SET TO 1
IPLOT =0 NO PLOTS; =1 PLOTS
IT(I) COORDINATE SYSTEM FOR BODY POINTS AT THIS STATION
 IT(I)=0 RECTANGULAR COORDINATES
 IT(I)=1 PCLAR CCOORDINATES
 IT(I)=2 PCLAR CCOORDINATES - AXISYMMERIC
ITYPE COORDINATE SYSTEM FOR STARTING PLANE INPUT
 ITYPE=0 PCLAR CCOORDINATES - AXISYMMERIC
 ITYPE=1 PCLAR COORDINATES
 ITYPE=2 RECTANGULAR COORDINATES
IT11 FLOW FIELD TAPE OUTPUT IT11=0 NO TAPE; IT11=1 TAPE
IT12 BODY DATA TAPE OUTPUT IT12=0 NO TAPE; IT12=1 TAPE
IV(I) NO. OF BODY PCINTS AT THIS STATION
JMAX MAXIMUM NO. OF PLANES TO BE COMPUTED
JPT PRINT EVERY JPT PLANES
NBT =0 INPUT FOR FIELD PCINT; =1 INPUT FOR BODY POINT
NE(J,K) THE J-TH NEIGHBOR OF THE K-TH POINT
NPTS TOTAL NO. OF POINTS IN STARTING PLANE
 IF ITYPE=0 NO. OF POINTS ON A RAY
NRAYS NO. OF RAYS IF ITYPE=0; OTHERWISE NOT USED
NS NO. OF STATIONS OF BODY GEOMETRY INPUT

INPUT NOMENCLATURE (CONT.)

P(I) PRESSURE
 PSI(I) ANGLE SHOWN IN FIG. 1; OR Y IF IMV=2
 PTO(I) TOTAL PRESSURE AT POINT (IF 0.0, SET TO PTO)
 PTO TOTAL PRESSURE FOR IRROTATIONAL FLOW
 Q(I) VELOCITY IF IMV=0; MACH NUMBER IF IMV=1; W IF IMV=2
 RO GAS CONSTANT
 THETA(I) ANGLE SHOWN IN FIG. 1; OR U IF IMV=2
 TTO TOTAL TEMPERATURE
 X(I) POINT IN STARTING PLANE
 X IF ITYPE=2; R IF ITYPE=0 OR 1
 XC(J,I) BODY POINT X IF IT(I)=0; R IF IT(I)=1 OR 2
 Y(I) POINT IN STARTING PLANE
 Y IF ITYPE=2; ANGLE IF ITYPE=0 OR 1
 YC(J,I) BODY POINT Y IF IT(I)=0; ANGLE IF IT(I)=1 OR 2
 ZC(I) Z LOCATION OF BODY STATION
 ZMAX MAXIMUM Z LOCATION FOR CALCULATIONS
 ZO Z LOCATION OF STARTING PLANE

INPUT CARDS IN ORDER READ

CARD	FORMAT	CONTENTS
1	(20A4)	(HEAD(I),I=1,20)
2	(80I1)	(IDBUG(I),I=1,80)
3	(I5)	NS
4	(2I5,E10.0)	IV(I),IT(I),ZC(I)
5	(8E10.0)	(XC(J,I),YC(J,I),J=1,IV(I))
READ CARDS 4 AND 5 NS TIMES		
6	(4E10.0)	G,PT0,TT0,R0
7	(2E10.0,10I5)	Z0,ZMAX,ITYPE,IMV,NPTS,NRAYS,JMAX, JPT,IPL0T,IT11,IT12,INEIGH
8	(7E10.0,I5)	X(I),Y(I),P(I),Q(I),THETA(I),PSI(I), PT0(I),NBP -- READ NPTS CARDS
9	(9I10)	(K,(NE(J,K),J=1,8),I=1,NPTS)
READ CARD 9 ONLY IF INEIGH=2		



```

C      THREE DIMENSIONAL DUCT FLOW                SRR00886
C      SIMPLIFIED VERSION WITH FOLLOWING ASSUMPTIONS
C      NO INTERNAL SHOCKS
C      IDEAL GAS
C      UNIT      FUNCTION
C      5          CARD INPUT
C      6          PRINTED OUTPUT
C      10         PLOTTED OUTPUT
C      11         FLOW FIELD OUTPUT
C      12         BODY OUTPUT
C      8          WORK DATA SET
C      OVERLAY STRUCTURE
C      OVERLAY A
C      INSERT CARDIN, INGEOM, INPUT, NEIGH, FIND, SORT
C      OVERLAY A
C      INSERT THREED, CUT, FIT, DIST, SING, DELTAF
C      OVERLAY B
C      INSERT FIELD
C      OVERLAY B
C      INSERT BODY, FINDBP, SOLVBP, NORMAL, BODYFIT, NEWRAP
C      OVERLAY B
C      INSERT OUTPUT
C      COMMON /TITLE/HEAD(20)
C      COMMON/IDBUG/IDBUG(80)
C      COMMON /STAG/DUMMY(1207), IPLIT, IT11, IT12
C      DATA ISTRP/0/
1  FORMAT(20A4)
2  FORMAT(80I1)
  CALL CARDIN
20 CONTINUE
  READ(8,1,END=99)HEAD
  READ(8,2)IDBUG
  CALL INGEOM
  CALL INPUT
  IF(IPLIT.EQ.0) GO TO 21
  IF(ISTRP.NE.0) GO TO 21
  ISTRP=1
  CALL IDENT(1)
21 CONTINUE
  CALL THREED
  GO TO 20
99 CONTINUE
  IF(ISTRP.NE.0) CALL IDENT(2)
  STOP
  END

```

```

SUBROUTINE BDYFIT(A,B,C,XB,YB,I,IT)
IMPLICIT REAL*8(A-H,O-Z)
REAL*4 DUMMY(5051),XB(50),YB(50)
COMMON /GEOM/DUMMY,NP
C      IT=1      X=C
C      IT=2      Y=C
C      IT=3      AX+BY=C
C      IT=4      (X-A)**2+(Y-B)**2=C

IM1=I-1
IF(I.EQ.1) IM1=NP
X1=XB(IM1)
Y1=YB(IM1)
X2=XB(I)
Y2=YB(I)
IP1=I+1
IF(I.EQ.NP) IP1=1
X3=XB(IP1)
Y3=YB(IP1)
IF((X1.NE.X2).OR.(X1.NE.X3)) GO TO 21
IT=1
A=1.0
B=0.0
C=X2
RETURN
21 CONTINUE
IF((Y1.NE.Y2).OR.(Y1.NE.Y3)) GO TO 22
IT=2
A=0.0
B=1.0
C=Y2
RETURN
22 CONTINUE
R1=X1**2+Y1**2
R2=X2**2+Y2**2
R3=X3**2+Y3**2
DX21=2.00*(X2-X1)
DY21=2.00*(Y2-Y1)
DX32=2.00*(X3-X2)
DY32=2.00*(Y3-Y2)
IF(DX21.NE.0.00) GO TO 23
DX21=2.00*(X3-X1)
DX32=-DX32
DY21=2.00*(Y3-Y1)
DY32=-DY32
H=R2
R2=R3
R3=H
GO TO 24
23 CONTINUE
IF(DX32.EQ.0.00) GO TO 24

```

```
      IF((DY21/DX21).NE.(DY32/DX32)) GO TO 24
      IT=3
      A=-DY21/DX21
      B=1.00
      C=Y1+X1*A
      RETURN
24  CONTINUE
      IT=4
      A=(R2-R1)/DX21
      B=R3-R2
      C=DY21/DX21
      H=DY32
      IF(DX32.EQ.0.00) GO TO 25
      B=B/DX32-A
      H=H/DX32-C
25  CONTINUE
      B=B/H
      A=A-B*C
      C=R1-2.00*(A*X1+B*Y1)
      C=C+B**2+A**2
      RETURN
      END
```

```

SUBROUTINE BODY(I)
  IMPLICIT REAL*8(A-H,O-Z)
  REAL*8 ZD,XN(1000),YN(1000),PN(1000),RN(1000),QN(1000),
  *PSN(1000),ZE,X(1000),Y(1000),P(1000),RHO(1000),
  *Q(1000),THETA(1000),XB(50),YB(50),XBO(50),YBO(50),
  *TN(1000),PSI(1000)
  COMMON /NVALU/ZD,XN,YN,PN,RN,QN,TN,PSN
  COMMON /VALUE/ZE,X,Y,P,RHO,Q,THETA,PSI,N
  COMMON /BCUT/XB ,YB
  COMMON /OCUT/XBO,YBO
  COMMON /DGAS/DZ,G,RO,GM1,GM1H,GM1G,GGM1,GP1,GPGM
  COMMON/IDBUG/IDBUG(80)
  REAL*8 N11,N21,N31,N12,N22,N32,N1,N2,N3
  COMMON /POINT/X1,Y1,P1,R1,Q1,T1,TH1,PS1,BE1,X2,Y2,P2,
  *R2,Q2,T2,TH2,PS2,BE2,P2O,T2O,PSO,OEL(4),PI(4),THI(4),
  *PII(4),BEI(4),XI(4),YI(4),DEI(4),DX(4),DY(4),DL(4),
  *PP(5,4),T12,P12,B12,T2I(4),P2I(4),B2I(4),D2I(4)
  COMMON /FLOBDY/SINT12,COST12,SINP12,COSP12,SINB12,
  *COSB12,SIND2I(4),COSD2I(4),SINT2I(4),COST2I(4),
  *SINP2I(4),COSP2I(4),SINB2I(4),COSB2I(4),DXDN(4),
  *DYDN(4),DZDN(4),DTEX(4),DTEY(4),DTEZ(4),DPDX(4),
  *DPDY(4),DPDZ(4),DER(2,5),AL(3,4),CR(4),AM(3,3),CM(3,4)
  1 FORMAT('0 FAILED IN BODY AT POINT',I5,5X,'Z1=',
  *IPE12,5,5X,'DZ=',E12,5//5X,'X1=',E12,5,5X,'Y1=',E12,5,
  *5X,'P1=',E12,5,5X,'T1=',E12,5,5X,'PS1=',E12,5/5X,'X2=',
  *,E12,5,5X,'Y2=',E12,5,5X,'P2=',E12,5,5X,'T2=',E12,5,5X
  *, 'PS2=',E12,5/45X,'P2O=',E12,5,4X,'T2O=',E12,5,4X,
  *'PS2O=',E12,5//)
  X1=X(I)
  Y1=Y(I)
  P1=P(I)
  R1=RHO(I)
  Q1=Q(I)
  TH1=THETA(I)
  PS1=PSI(I)
  T1=P1/(R0*R1)
  SINT1=DSIN(TH1)
  COST1=DCOS(TH1)
  SINP1=DSIN(PS1)
  COSP1=DCOS(PS1)
  BE1=DARSIN(DSORT(G*P1/R1)/Q1)
  P2=P1
  R2=R1
  Q2=Q1
  TH2=TH1
  PS2=PS1
  T2=T1
  BE2=BE1
  CALL FINDBP(X1,Y1,XBO,YBO,IBP1)
  CALL BDYFIT(A1,B1,C1,XBO,YBO,IBP1,IT1)

```

```

CALL BDYFIT(A2,B2,C2,XB,YB,IBP1,IT2)
CALL NORMAL(X1,Y1,ZE,A1,B1,C1,N11,N21,N31,IT1,IBP1)
CALL NORMAL(X1,Y1,ZE+DZ,A2,B2,C2,N12,N22,N32,IT2,IBP1)
DEL(1)=DARCOS(-N11*SINT1*COSP1+N21*COST1-N31*SINT1*
*SINP1)
IF(N31.LT.0.00) DEL(1)=6.28318530700-DEL(1)
DEL(2)=DEL(1)-1.57079632700
IF(DEL(2).LT.0.00) DEL(2)=6.28318530700+DEL(2)
DEL(3)=DEL(1)+1.57079632700
IF(DEL(3).GE.6.28318530700)
*DEL(3)=DEL(3)-6.28318530700
DO 21 IT=1,4
TH1(IT)=TH1
P11(IT)=PS1
BE1(IT)=BE1
DE1(IT)=DEL(IT)
21 CONTINUE
IBP2=IBP1
Y2=Y1
DO 25 IT=1,25
T12=0.500*(TH1+TH2)
P12=0.500*(PS1+PS2)
B12=0.500*(BE1+BE2)
SINT12=DSIN(T12)
COST12=DCOS(T12)
SINP12=DSIN(P12)
COSP12=DCOS(P12)
SINB12=DSIN(B12)
COSB12=DCOS(B12)
N1=0.500*(N11+N12)
N2=0.500*(N21+N22)
N3=0.500*(N31+N32)
CC1=N3*COST12*COSP12-N1*COST12*SINP12
CC2=N1*SINT12-N2*COST12*COSP12
CC3=(N3*SINT12-N2*COST12*SINP12)*DZ+CC1*X1+CC2*Y1
Y20=Y2
IBP=IBP2
X20=X2
CALL SOLVBP(X2,Y2,CC1,CC2,CC3,A2,B2,C2,IT2,X20,Y20)
CALL CKBDYP(X2,Y2,XB,YB,IBP2)
IF(IBP.NE.1BP2) CALL BDYFIT(A2,B2,C2,XB,YB,IBP2,IT2)
CALL NORMAL(X2,Y2,ZE+DZ,A2,B2,C2,N12,N22,N32,IT2,IBP2)
SINT2=DSIN(TH2)
COST2=DCOS(TH2)
SINP2=DSIN(PS2)
COSP2=DCOS(PS2)
DEL(1)=DARCOS(-N12*SINT2*COSP2+N22*COST2-N32*SINT2*
*SINP2)
IF(N32.LT.0.00) DEL(1)=6.28318530700-DEL(1)
DEL(2)=DEL(1)-1.57079632700

```

```

      IF(DEL(2).LT.0.00) DEL(2)=6.28318530700+DEL(2)
      DEL(3)=DEL(1)+1.57079632700
      IF(DEL(3).GT.6.28318530700)
      *DEL(3)=DEL(3)-6.28318530700
      DO 23 J=1,3
      T2I(J)=0.500*(TH2+THI(J))
      P2I(J)=0.500*(PS2+PII(J))
      B2I(J)=0.500*(BE2+BEI(J))
      D2I(J)=0.500*(DEL(J)+DEI(J))
      D2I(J)=D2I(J)-3.14159265400
      IF(D2I(J).LT.0.00) D2I(J)=6.28318530700+D2I(J)
      SINT2I(J)=DSIN(T2I(J))
      COST2I(J)=DCOS(T2I(J))
      SINP2I(J)=DSIN(P2I(J))
      COSP2I(J)=DCOS(P2I(J))
      SINB2I(J)=DSIN(B2I(J))
      COSB2I(J)=DCOS(B2I(J))
      SIND2I(J)=DSIN(D2I(J))
      COSD2I(J)=DCOS(D2I(J))
      F1=SINB2I(J)*COSD2I(J)
      F2=SINB2I(J)*SIND2I(J)
      F3=COSB2I(J)*COST2I(J)-F1*SINT2I(J)
      DL(J)=DZ/(F3*COSP2I(J)-F2*SINP2I(J))
      DX(J)=SINT2I(J)*COSB2I(J)+F1*COST2I(J)
      DY(J)=F3*SINP2I(J)+F2*COSP2I(J)
      XI(J)=X2-DX(J)*DL(J)
      YI(J)=Y2-DY(J)*DL(J)
      CALL GETPT(XI(J),YI(J),PP(1,J))
      RAD=DSQRT(G*PP(1,J)/PP(2,J))/PP(3,J)
      IF((RAD.GE.0.00).AND.(RAD.LE.1.00)) GO TO 22
      WRITE(6,2)I,J,P1,R1,Q1,TH1,PS1,BE1,P2,R2,Q2,TH2,PS2,
      *BE2,P20,R20,Q20,T20,PS0,DEL(1),DEL(2),DEL(3),N11,N21,
      *N31,N12,N22,N32,N1,N2,N3,DL(J),DX(J),DY(J),DEI(J),
      *PP(1,J),PP(2,J),PP(3,J),PP(4,J),PP(5,J),THI(J),PII(J),
      *BEI(J)
      2 FORMAT('1 FAILED IN BODY AT I=',I3.5X,'J=',I3//
      *1P6E14.5/6E14.5/5E14.5/3E14.5/9E14.5/4E14.5/8E14.5)
      STOP
22 CONTINUE
      CALL DELTAF(J,DEL(J),DEI(J))
      IF(DABS(DEI(J)).LE.1.0-8) DEI(J)=0.00
      CALL GETDER(XI(J),YI(J),DER)
      PI(J)=PP(1,J)
      THI(J)=PP(4,J)
      PII(J)=PP(5,J)
      BEI(J)=DARSIN(DSQRT(G*PP(1,J)/PP(2,J))/PP(3,J))
      T2I(J)=0.500*(TH2+THI(J))
      P2I(J)=0.500*(PS2+PII(J))
      B2I(J)=0.500*(BE2+BEI(J))
      D2I(J)=0.500*(DEL(J)+DEI(J))

```

```

D2I(J)=D2I(J)-3.141592654*D0
IF(D2I(J).LT.0.D0) D2I(J)=6.283185307*D0+D2I(J)
SINT2I(J)=DSIN(T2I(J))
COST2I(J)=DCOS(T2I(J))
SINP2I(J)=DSIN(P2I(J))
COSP2I(J)=DCOS(P2I(J))
SINB2I(J)=DSIN(B2I(J))
COSB2I(J)=DCOS(B2I(J))
SIND2I(J)=DSIN(D2I(J))
COSD2I(J)=DCOS(D2I(J))
OTDX(J)=DER(1,4)
DPDX(J)=DER(1,5)
OTDY(J)=DER(2,4)
DPDY(J)=DER(2,5)
F1=COST2I(J)*SIND2I(J)
F2=SINT2I(J)*SIND2I(J)
F3=SINB2I(J)*DL(J)
F4=COST2I(J)*COSD2I(J)
DXDN(J)=-F1
DYDN(J)=F2*SINP2I(J)+COSP2I(J)*COSD2I(J)
DZDN(J)=F2*COSP2I(J)-SINP2I(J)*COSD2I(J)
F5=XI(J)-X2
F6=YI(J)-Y2
DTDZ(J)=(OTDX(J)*F5+OTDY(J)*F6)/DZ
DPDZ(J)=(DPDX(J)*F5+DPDY(J)*F6)/DZ
AL(1,J)=COSB2I(J)/(SINB2I(J)*PP(2,J)*PP(3,J)**2)
AL(2,J)=COSD2I(J)-F3*SIND2I(J)*DZDN(J)/DZ
AL(3,J)=F1+F3*F4*DZDN(J)/DZ
CR(J)=F3*(SIND2I(J)*(OTDX(J)*DXDN(J)+OTDY(J)*DYDN(J)+
*DTDZ(J)*DZDN(J))-F4*(DPDX(J)*DXDN(J)+DPDY(J)*DYDN(J)+
*DPDZ(J)*DZDN(J))+AL(1,J)*PI(J)+AL(2,J)*THI(J)+AL(3,J)
*PII(J)
23 CONTINUE
A1=AL(2,1)*AL(1,2)-AL(1,1)*AL(2,2)
B1=AL(3,1)*AL(1,2)-AL(1,1)*AL(3,2)
C1=CR(1)*AL(1,2)-AL(1,1)*CR(2)
AM(2,1)=TH2
AM(3,1)=PS2
CALL NEWRAP(AM(2,1),AM(3,1),A1,B1,C1,N12,N22,N32)
A1=AL(2,2)*AL(1,3)-AL(1,2)*AL(2,3)
B1=AL(3,2)*AL(1,3)-AL(1,2)*AL(3,3)
C1=CR(2)*AL(1,3)-AL(1,2)*CR(3)
AM(2,2)=TH2
AM(3,2)=PS2
CALL NEWRAP(AM(2,2),AM(3,2),A1,B1,C1,N12,N22,N32)
A1=AL(2,3)*AL(1,1)-AL(1,3)*AL(2,1)
B1=AL(3,3)*AL(1,1)-AL(1,3)*AL(3,1)
C1=CR(3)*AL(1,1)-AL(1,3)*CR(1)
AM(2,3)=TH2
AM(3,3)=PS2

```

```

CALL NEWRAP(AM(2,3),AM(3,3),A1,B1,C1,N12,N22,N32)
AM(1,1)=(CR(1)-AL(2,1)*AM(2,1)-AL(3,1)*AM(3,1))/AL(1,1)
AM(1,2)=(CR(2)-AL(2,2)*AM(2,2)-AL(3,2)*AM(3,2))/AL(1,2)
AM(1,3)=(CR(3)-AL(2,3)*AM(2,3)-AL(3,3)*AM(3,3))/AL(1,3)
P2=0.500*(AM(1,1)+AM(1,3))
TH2=0.500*(AM(2,1)+AM(2,3))
PS2=0.500*(AM(3,1)+AM(3,3))
T2=TTOD*(P2/PTOD(I))*GGM1G
R2=P2/(R0*T2)
Q2=DSQRT(2.00*GGM1*R0*(TTOD-T2))
IF(IT.EQ.1) GO TO 24
IF(DABS((P2-P20)/P2).LE.1.0-7) GO TO 26
IF(IT.EQ.25) GO TO 25
24 CONTINUE
P20=P2
R20=R2
Q20=Q2
T20=TH2
PS0=PS2
BE2=DARSIN(DSQRT(G*P2/R2)/Q2)
25 CONTINUE
IF(DABS((P2-P20)/P2).LE.1.0-5) GO TO 26
WRITE(6,1)I,ZE,DZ,X1,Y1,P1,TH1,PS1,X2,Y2,P2,TH2,PS2,
*P20,T20,PS0
IF(DABS((P2-P20)/P2).GT.1.0-4) STOP
26 CONTINUE
XN(I)=X2
YN(I)=Y2
PN(I)=P2
RN(I)=R2
QN(I)=Q2
TN(I)=TH2
PSN(I)=PS2
RETURN
END

```

```
      SUBROUTINE CARDIN
      DIMENSION A(20)
      1 FORMAT(20A4)
      2 FORMAT(5X,20A4)
      3 FORMAT('1',38X,'INPUT CARDS')
      4 FORMAT('0',4X,9('0'),10('1'),10('2'),10('3'),10('4'),
      *10('5'),10('6'),10('7'),'8'/5X,8('1234567890')/)
      IP=50
      WRITE(6,3)
      WRITE(6,4)
20  CONTINUE
      READ(5,1,END=99)A
      IF(IP.NE.0) GO TO 21
      WRITE(6,4)
      WRITE(6,3)
      WRITE(6,4)
      IP=50
21  CONTINUE
      WRITE(6,2)A
      WRITE(8,1)A
      IP=IP-1
      GO TO 20
99  CONTINUE
      REWIND 8
      WRITE(6,4)
      RETURN
      END
```

```

SUBROUTINE CUT(ZB)
COMMON /BCUT/XB(50),YB(50)
COMMON /GEOM/ZC(50),XC(50,50),YC(50,50),NS,NP
REAL*4 A(3,3)
Z=ZB
DO 21 I=2,NS
  IF(ZB.LE.ZC(I)) GO TO 22
21 CONTINUE
  I=NS-1
22 CONTINUE
  IF(I.EQ.NS) I=NS-1
  IP=I+1
  IM=I-1
  ZO=ZC(IM)
  ZI=ZC(I)
  Z2=ZC(IP)
  Z10=1.0/(ZI-ZO)
  Z20= 1.0/(Z2-ZO)
  Z21=1.0/(Z2-ZI)
  ZP1=ZO+ZI
  A(3,1)=Z21*(Z10-Z20)
  A(3,2)=-Z21*Z10
  A(3,3)= Z21*Z20
  A(2,1)=-Z10-ZP1*A(3,1)
  A(2,2)= Z10-ZP1*A(3,2)
  A(2,3)= -ZP1*A(3,3)
  A(1,1)=-ZO*(A(2,1)+A(3,1)*ZO)+1.0
  A(1,2)=-ZO*(A(2,2)+A(3,2)*ZO)
  A(1,3)=-ZO*(A(2,3)+A(3,3)*ZO)
  DO 23 J=1,NP
    B=A(1,1)*XC(J,IM)+A(1,2)*XC(J,I)+A(1,3)*XC(J,IP)
    C=A(2,1)*XC(J,IM)+A(2,2)*XC(J,I)+A(2,3)*XC(J,IP)
    D=A(3,1)*XC(J,IM)+A(3,2)*XC(J,I)+A(3,3)*XC(J,IP)
    XB(J)=B+(C+D*Z)*Z
    B=A(1,1)*YC(J,IM)+A(1,2)*YC(J,I)+A(1,3)*YC(J,IP)
    C=A(2,1)*YC(J,IM)+A(2,2)*YC(J,I)+A(2,3)*YC(J,IP)
    D=A(3,1)*YC(J,IM)+A(3,2)*YC(J,I)+A(3,3)*YC(J,IP)
    YB(J)=B+(C+D*Z)*Z
23 CONTINUE
  RETURN
END

```

```

SUBROUTINE DELTAF(J,DEL,DELI)
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /FLDBDY/DUMMY(14),SINTI(4),COSTI(4),SINPI(4),
  *COSPI(4),SINBI(4),COSBI(4)
  A=SINTI(J)*COSPI(J)*SINBI(J)
  B=SINPI(J)*SINBI(J)
  C=-0.500*COSTI(J)*COSPI(J)*(SINBI(J)+COSBI(J))
  SIND=DSIN(DEL)
  COSD=DCOS(DEL)
  DELI=DEL
  ASBS=A**2+B**2
  RAD=ASBS-C**2
  IF(RAD.LT.0.00) RETURN
  RAD=DSQRT(RAD)/ASBS
  SINI=B*C/ASBS
  SINI1=SINI+A*RAD
  SINI2=SINI-A*RAD
  IF(DABS(SIND-SINI2).LT.DABS(SIND-SINI1)) GO TO 21
  SINI=SINI1
  IF(DABS(SINI).LE.1.00) GO TO 22
21 CONTINUE
  SINI=SINI2
  IF(DABS(SINI).GT.1.00) SINI=SINI1
  IF(DABS(SINI).GT.1.00) RETURN
22 CONTINUE
  COSI=A*C/ASBS
  COSI1=COSI+B*RAD
  COSI2=COSI-B*RAD
  IF(DABS(COSD-COSI2).LT.DABS(COSD-COSI1)) GO TO 23
  COSI=COSI1
  IF(DABS(COSI).LE.1.00) GO TO 24
23 CONTINUE
  COSI=COSI2
  IF(DABS(COSI).GT.1.00) COSI=COSI1
  IF(DABS(COSI).GT.1.00) RETURN
24 CONTINUE
  DELI=DATAN2(SINI,COSI)
  IF(DELI.LT.0.00) DELI=6.28318530700+DELI
  RETURN
END

```

```

SUBROUTINE DIST(DELS)
COMMON /GAS/G,RO,GM1,GM1H,GM1G,GGM1,GP1,GPGM
COMMON/VALUE/ZE,X(1000),Y(1000),P(1000),RHO(1000) ,
*Q(1000),THETA(1000),PSI(1000),N
INTEGER*2 NE(8,1000)
COMMON /NEAR/NE,DMIN
DMIN=1.E10
QMIN=1.E10
AMAX=0.0
DO 22 I=1,N
A=P(I)/RHO(I)
IF(A.GT.AMAX) AMAX=A
IF(Q(I).LT.QMIN) QMIN=Q(I)
DO 21 J=1,8
K=NE(J,I)
D=(X(I)-X(K))**2+(Y(I)-Y(K))**2
IF(D.LT.DMIN) DMIN=D
21 CONTINUE
22 CONTINUE
AMAX=SQRT(G*AMAX)
DELS=QMIN*SQRT(DMIN)/AMAX
DELS=SQRT(DMIN*ABS((QMIN/AMAX)**2-1.0))
RETURN
END

```

```

SUBROUTINE FIELD(I)
  IMPLICIT REAL*8(A-H,O-Z)
  REAL*4 ZD,XN(1000),YN(1000),PN(1000),RN(1000),QN(1000),
  *PSN(1000),ZE,X(1000),Y(1000),P(1000),RHO(1000),
  *TN(1000),PSI(1000),Q(1000),THETA(1000)
  COMMON /NVALU/ZD,XN,YN,PN,RN,QN,TN,PSN
  COMMON /VALUE/ZE,X,Y,P,RHO,Q,THETA,PSI,N
  COMMON /DGAS/DZ,G,RO,GM1,GM1H,GM1G,GGM1,GP1,GPGM
  COMMON/DSTAG/PTOD(1000),TTOD,ATOD
  COMMON/IDEBUG/IDBUG(80)
  COMMON /POINT/X1,Y1,P1,R1,Q1,T1,TH1,PS1,BE1,X2,Y2,P2,
  *R2,Q2,T2,TH2,PS2,BE2,P2O,T2O,PSO,DEL(4),P1(4),TH1(4),
  *P11(4),BE1(4),X1(4),Y1(4),DE1(4),DX(4),DY(4),DL(4),
  *PP(5,4),T12,P12,B12,T21(4),P21(4),B21(4),D21(4)
  COMMON /FLDBDY/SINT12,COST12,SINP12,COSP12,SINB12,
  *COSB12,SIND21(4),COSD21(4),SINT21(4),COST21(4),
  *SINP21(4),COSP21(4),SINB21(4),COSB21(4),DXDN(4),
  *DYDN(4),DZDN(4),DTDX(4),DTDY(4),DTDZ(4),DPDX(4),
  *DPDY(4),DPDZ(4),DER(2,5),AL(3,4),CR(4),AM(3,3),CM(3,4)
1  FORMAT('0  FAILED IN FIELD AT POINT',I5,5X,'Z1=',
  *IPE12,5,5X,'DZ=',E12,5//5X,'X1=',E12,5,5X,'Y1=',E12,5,
  *5X,'P1=',E12,5,5X,'T1=',E12,5,5X,'PS1=',E12,5//5X,'X2=',
  *,E12,5,5X,'Y2=',E12,5,5X,'P2=',E12,5,5X,'T2=',E12,5,5X
  *,',PS2=',E12,5//45X,'P2O=',E12,5,4X,'T2O=',E12,5,4X,
  *',PS2O=',E12,5//)
  X1=X(I)
  Y1=Y(I)
  DEL(1)=0.D0
  IF(X1.NE.0.D0.OR.Y1.NE.0.D0) DEL(1)=DATAN2(Y1,X1)
  IF(DEL(1).LT.0.D0) DEL(1)=6.28318530700+DEL(1)
  DE1(1)=DEL(1)
  DO 20 IT=2,4
  DEL(IT)=DEL(IT-1)+1.57079632700
  IF(DEL(IT).GE.6.28318530700)
  *DEL(IT)=DEL(IT)-6.28318530700
  DE1(IT)=DEL(IT)
20 CONTINUE
  P1=P(I)
  R1=RHO(I)
  Q1=Q(I)
  TH1=THETA(I)
  PS1=PSI(I)
  T1=P1/(R0*R1)
  P2=P1
  R2=R1
  Q2=Q1
  TH2=TH1
  PS2=PS1
  T2=T1
  BE1=DARSIN(DSQRT(G*P1/R1)/Q1)

```

```

BE2=BE1
DO 21 IT=1,4
  TH1(IT)=TH1
  P11(IT)=PS1
  BE1(IT)=BE1
21 CONTINUE
DO 26 IT=1,25
  T12=0.5D0*(TH1+TH2)
  P12=0.5D0*(PS1+PS2)
  B12=0.5D0*(BE1+BE2)
  SINT12=DSIN(T12)
  COST12=DCOS(T12)
  SINP12=DSIN(P12)
  COSP12=DCOS(P12)
  SINB12=DSIN(B12)
  COSB12=DCOS(B12)
  X2=X1+SINT12*DZ/(COST12*COSP12)
  Y2=Y1+SINP12*DZ/COSP12
DO 22 J=1,4
  T2I(J)=0.5D0*(TH2+TH1(J))
  P2I(J)=0.5D0*(PS2+P1I(J))
  B2I(J)=0.5D0*(BE2+BE1(J))
  D2I(J)=0.5D0*(DEL(J)+DEI(J))
  SINT2I(J)=DSIN(T2I(J))
  COST2I(J)=DCOS(T2I(J))
  SINP2I(J)=DSIN(P2I(J))
  COSP2I(J)=DCOS(P2I(J))
  SINB2I(J)=DSIN(B2I(J))
  COSB2I(J)=DCOS(B2I(J))
  SIND2I(J)=DSIN(D2I(J))
  COSD2I(J)=DCOS(D2I(J))
  F1=SINB2I(J)*COSD2I(J)
  F2=SINB2I(J)*SIND2I(J)
  F3=COSB2I(J)*COST2I(J)-F1*SINT2I(J)
  DL(J)=DZ/(F3*COSP2I(J)-F2*SINP2I(J))
  DX(J)=SINT2I(J)*COSB2I(J)+F1*COST2I(J)
  DY(J)=F3*SINP2I(J)+F2*COSP2I(J)
  XI(J)=X2-DX(J)*DL(J)
  YI(J)=Y2-DY(J)*DL(J)
  CALL GETPT(XI(J),YI(J),PP(1,J))
  IF(DABS(PP(4,J)).LT.1.D-8) PP(4,J)=0.00
  IF(DABS(PP(5,J)).LT.1.D-8) PP(5,J)=0.00
  CALL DELTAF(J,DEL(J),DEI(J))
  IF(DABS(DEI(J)).LE.1.0-8) DEI(J)=0.00
  CALL GETDER(XI(J),YI(J),DER)
  P1(J)=PP(1,J)
  TH1(J)=PP(4,J)
  P11(J)=PP(5,J)
  BE1(J)=DARSIN(DSORT(G*PP(1,J)/PP(2,J))/PP(3,J))
  T2I(J)=0.5D0*(TH2+TH1(J))

```

```

P2I(J)=0.5D0*(PS2+PII(J))
B2I(J)=0.5D0*(BE2+BEI(J))
D2I(J)=0.5D0*(DEL(J)+DEI(J))
SINT2I(J)=DSIN(T2I(J))
COST2I(J)=DCOS(T2I(J))
SINP2I(J)=DSIN(P2I(J))
COSP2I(J)=DCOS(P2I(J))
SINB2I(J)=DSIN(B2I(J))
COSB2I(J)=DCOS(B2I(J))
SIND2I(J)=DSIN(D2I(J))
COSD2I(J)=DCOS(D2I(J))
DTDX(J)=DER(1,4)
DPDX(J)=DER(1,5)
DTDY(J)=DER(2,4)
DPDY(J)=DER(2,5)
F1=COST2I(J)*SIND2I(J)
F2=SINT2I(J)*SIND2I(J)
F3=SINB2I(J)*DL(J)
F4=COST2I(J)*COSD2I(J)
DXDN(J)=-F1
DYDN(J)=F2*SINP2I(J)+COSP2I(J)*COSD2I(J)
DZDN(J)=F2*COSP2I(J)-SINP2I(J)*COSD2I(J)
F5=XI(J)-X2
F6=YI(J)-Y2
DTDZ(J)=(DTDX(J)*F5+DTDY(J)*F6)/DZ
DPDZ(J)=(DPDX(J)*F5+DPDY(J)*F6)/DZ
AL(1,J)=COSB2I(J)/(SINB2I(J)*PP(2,J)*PP(3,J)**2)
AL(2,J)=COSD2I(J)-F3*SIND2I(J)*DZDN(J)/DZ
AL(3,J)=F1+F3*F4*DZDN(J)/DZ
CR(J)=F3*(SIND2I(J)*(DTDX(J)*DXDN(J)+DTDY(J)*DYDN(J)+
*DTDZ(J)*DZDN(J))-F4*(DPDX(J)*DXDN(J)+DPDY(J)*DYDN(J)+
*DPDZ(J)*DZDN(J))+AL(1,J)*PI(J)+AL(2,J)*THI(J)+AL(3,J)
**PII(J)

```

22 CONTINUE

DO 24 M=1,4

L=M

DO 23 J=1,3

AM(J,1)=AL(1,L)

AM(J,2)=AL(2,L)

AM(J,3)=AL(3,L)

CM(J,M)=CR(L)

L=L+1

IF(L.GT.4) L=1

23 CONTINUE

CALL SIMQ(AM,CM(1,M),3,1)

24 CONTINUE

P2 =0.25D0*(CM(1,1)+CM(1,2)+CM(1,3)+CM(1,4))

TH2=0.25D0*(CM(2,1)+CM(2,2)+CM(2,3)+CM(2,4))

PS2=0.25D0*(CM(3,1)+CM(3,2)+CM(3,3)+CM(3,4))

T2=TTOD*(P2/PTOD(I))**GMIG

```

R2=P2/(R0*T2)
Q2=DSQRT(2.00*GGM1*R0*(TT00-T2))
IF(IT.EQ.1) GO TO 25
IF(DABS((P2-P20)/P2).LE.1.D-7) GO TO 27
IF(IT.EQ.25) GO TO 26
25 CONTINUE
P20=P2
R20=R2
Q20=Q2
T20=TH2
PS0=PS2
BE2=DARSIN(DSQRT(G*P2/R2)/Q2)
26 CONTINUE
IF(DABS((P2-P20)/P2).LE.1.D-5) GO TO 27
WRITE(6,1)I,ZE,OZ,X1,Y1,P1,TH1,PS1,X2,Y2,P2,TH2,PS2,
*P20,T20,PS0
IF(DABS((P2-P20)/P2).GT.1.D-4) STOP
27 CONTINUE
XN(I)=X2
YN(I)=Y2
PN(I)=P2
RN(I)=R2
QN(I)=Q2
TN(I)=TH2
PSN(I)=PS2
RETURN
END

```

```

SUBROUTINE FIND(I,L1,L3,NE)
  INTEGER*2 NE(1)
  REAL*4 DIST(3)
  COMMON/VALUE/ZE,X(1000),Y(1000)
  L2=L1+L3
  J=0
  DO 22 L=L1,L2
    D=(X(I)-X(L))*2+(Y(I)-Y(L))*2
    IF(J.EQ.3) GO TO 21
    J=J+1
    NE(J)=L
    DIST(J)=D
    IF(J.NE.3) GO TO 22
    CALL SORT(DIST,NE,3)
    GO TO 22
21 CONTINUE
    IF(D.GE.DIST(3)) GO TO 22
    DIST(3)=D
    NE(3)=L
    CALL SORT(DIST,NE,3)
22 CONTINUE
  RETURN
  END

```

```

SUBROUTINE FINDBP(X1,Y1,XB,YB,IPOINT)
COMMON /GEOM/ZC(50),XC(50,50),YC(50,50),NS,NP
DIMENSION XB(50),YB(50)
20 CONTINUE
IPOINT=1
DMIN=(X1-XB(1))**2+(Y1-YB(1))**2
DO 21 J=2,NP
D=(X1-XB(J))**2+(Y1-YB(J))**2
IF(D.GE.DMIN) GO TO 21
IPOINT=J
DMIN=D
21 CONTINUE
RETURN
ENTRY CKBDYP(X1,Y1,XB,YB,IPOINT)
J=IPOINT
DMIN=(X1-XB(J))**2+(Y1-YB(J))**2
J=J-1
IF(J.EQ.0) J=NP
D=(X1-XB(J))**2+(Y1-YB(J))**2
IF(D.LT.DMIN) GO TO 20
J=IPOINT+1
IF(J.GT.NP) J=1
D=(X1-XB(J))**2+(Y1-YB(J))**2
IF(D.LT.DMIN) GO TO 20
RETURN
END

```

```

SUBROUTINE FIT(I)
COMMON/VALUE/ZE,X(1000),Y(1000),P(1000),RHO(1000) ,
*Q(1000),THETA(1000),PSI(1000),N
INTEGER*2 NE(8,1000)
COMMON /NEAR /NE
REAL*8 R,B(12,12),C(12,5),VAL(5),DER(2,5),DX,DY,FAC
DO 22 J=1,12
IF(J.GT.9) GO TO 21
B(1,J)=1.00
K=1
IF(J.NE.1) K=NE(J-1,1)
B(2,J)=X(K)
B(3,J)=Y(K)
GO TO 22
21 CONTINUE
B(1,J)=0.00
B(2,J)=0.00
B(3,J)=0.00
22 CONTINUE
DO 24 JJ=4,11
J=13-JJ
KL=J-1
K1=1
IF(J.NE.1) K1=NE(KL,1)
DO 23 K=1,KL
K2=1
IF(K.NE.1) K2=NE(K-1,1)
R=(X(K1)-X(K2))**2+(Y(K1)-Y(K2))**2
B(JJ,K)=R*DLOG(R)
23 CONTINUE
B(JJ,J)=0.00
24 CONTINUE
B(12,1)=0.00
DO 26 J=1,11
K=13-J
KL=K-1
DO 25 L=1,KL
B(K,13-L)=B(L,J)
25 CONTINUE
26 CONTINUE
DO 29 J=1,12
IF(J.GT.3) GO TO 28
DO 27 K=1,5
C(J,K)=0.00
27 CONTINUE
GO TO 29
28 CONTINUE
KL=13-J
K=1
IF(KL.NE.1) K=NE(KL-1,1)

```

```

      C(J,1)=P(K)
      C(J,2)=RHO(K)
      C(J,3)=Q(K)
      C(J,4)=THETA(K)
      C(J,5)=PSI(K)
29  CONTINUE
      CALL SIMQ(B,C,12,5)
      RETURN
      ENTRY GETPT(X1,Y1,VAL)
      DO 30 J=1,5
        VAL(J)=C(12,J)+C(11,J)*X1+C(10,J)*Y1
30  CONTINUE
      DO 32 J=1,9
        K=1
        IF(J.NE.1) K=NE(J-1,1)
        R=(X1-X(K))**2+(Y1-Y(K))**2
        IF(R.EQ.0.D0) GO TO 32
        DO 31 L=1,5
          VAL(L)=VAL(L)+C(J,L)*R*DLOG(R)
31  CONTINUE
32  CONTINUE
      RETURN
      ENTRY GETDER(X1,X2,DER)
      DO 33 J=1,5
        DER(1,J)=C(11,J)
        DER(2,J)=C(10,J)
33  CONTINUE
      DO 35 J=1,9
        K=1
        IF(J.NE.1) K=NE(J-1,1)
        DX=X1-X(K)
        DY=Y1-Y(K)
        R=DX**2+DY**2
        IF(R.EQ.0.D0) GO TO 35
        DO 34 L=1,5
          FAC=2.D0*C(J,L)*(1.D0+DLOG(R))
          DER(1,L)=DER(1,L)+FAC*DX
          DER(2,L)=DER(2,L)+FAC*DY
34  CONTINUE
35  CONTINUE
      RETURN
      END

```

```
SUBROUTINE IDENT(I)
  IF(I.EQ.2) GO TO 21
  CALL CALCMP(15.0,2.0,10.0)
  CALL SYMBOL(0.5,7.5,0.84,'FOLLOWING PLOTS FOR',0.0,19)
  CALL SYMBOL(1.2,5.5,0.98,'W. C. ARMSTRONG',0.0,15)
  CALL SYMBOL(5.5,3.5,0.84,'CSB-EAD',0.0,7)
  CALL CLASS(1,3,0,0)
  CALL CALCMP(0.5,0.5,0.3)
  RETURN
21 CONTINUE
  CALL CALCMP(0.0,0.0,0.3)
  CALL CLASS(4,3,0,0)
  CALL SYMBOL(.08,7.5,0.84,'PRECEEDING PLOTS FOR',0.0,20)
  CALL SYMBOL(1.2,5.5,0.98,'W. C. ARMSTRONG',0.0,15)
  CALL SYMBOL(5.5,3.5,0.84,'CSB-EAD',0.0,7)
  CALL CALCMP(0.0,0.0,9999.2)
  RETURN
END
```

```

SUBROUTINE INGEOM
COMMON /TITLE/HEAD(20)
COMMON /GEOM/ZC(50),XC(50,50),YC(50,50),NS,NP
COMMON/IDBUG/IDBUG(80)
COMMON/VALUE/X(50,50),Y(50,50),S(50),IT(50),IV(50)
1 FORMAT('1',10X,20A4//20X,'GEOMETRY INPUT',10X,
  *'NO. OF STATIONS =' ,I3/)
2 FORMAT('0  Z =' ,1PE13.5,5X,'IV =' ,I3,5X,'IT =' ,I2//
  *8X,'X',14X,'Y'//)
3 FORMAT(1P2E15.5)
4 FORMAT('1',10X,20A4//20X,'CONVERTED GEOMETRY'//)
5 FORMAT(2I5,E10.0)
6 FORMAT(8E10.0)
C      IV= NO. OF POINTS AT THIS STATION
C      IT= 1 - ALL POINTS GIVEN IN RECTANGULAR COORDINATES
C          2 - ALL POINTS GIVEN IN POLAR COORDINATES
C          3 - ONE POINT GIVEN IN POLAR COORDINATES
      READ(8,5)NS
      IF(IDBUG( 1),NE.0)
        *WRITE(6,1)HEAD,NS
        DO 21 I=1,NS
          READ(8,5)IV(I),IT(I),ZC(I)
          IF(IDBUG( 1),NE.0)
            *WRITE(6,2)ZC(I),IV(I),IT(I)
            N=IV(I)
            IF(IT(I).EQ.3) N=1
            READ(8,6)(X(J,I),Y(J,I),J=1,N)
            IF(IDBUG( 1),NE.0)
              *WRITE(6,3)(X(J,I),Y(J,I),J=1,N)
21 CONTINUE
          DO 26 I=1,NS
            IF(IDBUG( 2),NE.0)
              *WRITE(6,4)HEAD
              IF(IDBUG( 2),NE.0)
                *WRITE(6,2)ZC(I),IV(I),IT(I)
                N=IV(I)
                IF(IT(I).EQ.1) GO TO 25
                IF(IT(I).EQ.2) GO TO 23
                DT=360.0/N
                Y(1,I)=0.0
                DO 22 J=2,N
                  X(J,I)=X(1,I)
                  Y(J,I)=(J-1)*DT
22 CONTINUE
23 CONTINUE
                DO 24 J=1,N
                  XX=X(J,I)*COS(0.01745329*Y(J,I))
                  Y(J,I)=X(J,I)*SIN(0.01745329*Y(J,I))
                  X(J,I)=XX
24 CONTINUE

```

```

25 CONTINUE
  IF(IDBUG( 2).NE.0)
    *WRITE(6,3)(X(J,I),Y(J,I),J=1,N)
26 CONTINUE
  NP=IV(1)
  DO 27 I=2,NS
    IF(NP.GE.IV(I)) GO TO 27
    NP=IV(I)
27 CONTINUE
  NPM=NP-1
  FAC=1.0/NPM
  DO 33 I=1,NS
    N=IV(I)
    S(1)=0.0
    DO 28 J=2,N
      S(J)=S(J-1)+SQRT((X(J,I)-X(J-1,I))**2+
        *(Y(J,I)-Y(J-1,I))**2)
28 CONTINUE
    DS=S(N)*FAC
    XC(1,I)=X(1,I)
    YC(1,I)=Y(1,I)
    XC(NP,I)=X(N,I)
    YC(NP,I)=Y(N,I)
    KL=2
    DO 32 J=2,NPM
      SP=DS*(J-1)
      DO 29 K=KL,N
        IF(SP-S(K))31,30,29
29 CONTINUE
      K=N
30 CONTINUE
      XC(J,I)=X(K,I)
      YC(J,I)=Y(K,I)
      KL=K
      GO TO 32
31 CONTINUE
      RAT=(SP-S(K-1))/(S(K)-S(K-1))
      XC(J,I)=X(K-1,I)+(X(K,I)-X(K-1,I))*RAT
      YC(J,I)=Y(K-1,I)+(Y(K,I)-Y(K-1,I))*RAT
      KL=K
32 CONTINUE
33 CONTINUE
  RETURN
END

```

```

SUBROUTINE INPUT
COMMON /TITLE/HEAD(20)
COMMON/VALUE/ZE,X(1000),Y(1000),P(1000),RHO(1000) ,
*Q(1000),THETA(1000),PSI(1000),N
COMMON /STAG/PTO(1000),TTC,ATOT,ZO,ZMAX,NB(200),
*NBO(200),JMAX,JPT,IB,IPLT,IT11,IT12
COMMON /GAS/G,RO,GM1,GM1H,GM1G,GGM1,GP1,GPGM
COMMON/IDBUG/IDBUG(80)
INTEGER*2 NB,NBO
COMMON/NVALU/XR(20),YR(20),PR(20),THR(20),PSR(20),
*QR(20),PTR(20)
1 FORMAT('1',10X,20A4//20X,'STARTING PLANE INPUT'//5X,
*'GAMMA =' ,F5.3,5X,'PTO =' ,1PE12.5,5X,'TTC =' ,E12.5,
*5X,'R =' ,E12.5)
2 FORMAT('0 ZO=' ,1PE12.5,5X,'ZMAX=' ,E12.5,5X,' ITYPE=' ,
*I2,5X,'IMV=' ,I2,5X,'NPTS=' ,I5,5X,'NRAYS=' ,I3,5X,
*'JMAX=' ,I5,5X,'JPT=' ,I5/44X,'IPLT=' ,I2,4X,' IT11=' ,I2,
*5X,'IT12=' ,I2,7X,'INEIGH=' ,I2//8X,'X' ,14X,'Y' ,14X,'P' ,
*14X,'Q' ,12X,'THETA' ,11X,'PSI' ,12X,'PTO'//)
3 FORMAT(1P7E15.5)
4 FORMAT(7E10.0,I5)
5 FORMAT(2E10.0,I0I5)
6 FORMAT('1',10X,20A4//20X,'STARTING PLANE VALUES'//8X,
*'X' ,14X,'Y' ,14X,'P' ,13X,'RHO' ,13X,'Q' ,13X,'THR' ,12X,
*'PSI' ,12X,'PTO'//(1P8E15.5))
7 FORMAT('0 TOO MANY BODY POINTS')
C      IMV=0      Q(I)=Q      THETA(I)=THETA(I)      PSI(I)=PSI
C      IMV=1      Q(I)=M      THETA(I)=THETA      PSI(I)=PSI
C      IMV=2      Q(I)=W      THETA(I)=U      PSI(I)=V
C      ITYPE=0      ONE RAY GIVEN IN POLAR COORDINATES
C      ITYPE=1      PLANE GIVEN IN POLAR COORDINATES
C      ITYPE=2      PLANE GIVEN IN RECTANGULAR COORDINATES
READ(8,4)G,PTO,TTC,RO
WRITE(6,1)HEAD,G,PTO,TTC,RO
READ(8,5)ZO,ZMAX,ITYPE,IMV,NPTS,NRAYS,JMAX,JPT,IPLT,
*IT11,IT12,INEIGH
WRITE(6,2)ZO,ZMAX,ITYPE,IMV,NPTS,NRAYS,JMAX,JPT,IPLT,
*IT11,IT12,INEIGH
N=NPTS
IB=0
DO 20 I=1,N
READ(8,4)X(I),Y(I),P(I),Q(I),THETA(I),PSI(I),PTO(I),NBP
IF(PTO(I).EQ.0.0) PTO(I)=PTO
IF(IMV.EQ.2) GO TO 19
THETA(I)=0.01745329*THETA(I)
PSI(I)=0.01745329*PSI(I)
19 CONTINUE
IF(NBP.EQ.0) GO TO 20
IB=IB+1
IF(IB.GT.200) GO TO 99

```

```

      NB(IB)=I
20  CONTINUE
      IF(IDBUG( 3).NE.0)
        *WRITE(6,3)(X(I),Y(I),P(I),Q(I),THETA(I),PSI(I),
        *PTO(I),I=1,N)
      IF(ITYPE.EQ.1) GO TO 24
      IF(ITYPE.EQ.2) GO TO 26
      DO 21 I=1,N
        XR(I)=X(I)
        YR(I)=Y(I)
        PR(I)=P(I)
        QR(I)=Q(I)
        THR(I)=THETA(I)
        PSR(I)=PSI(I)
        PTR(I)=PTO(I)
21  CONTINUE
      IB=1
      J=0
      NBI=NB(1)
      DO 23 I=1,N
        J=J+1
        X(J)=XR(I)
        Y(J)=YR(I)
        P(J)=PR(I)
        Q(J)=QR(I)
        THETA(J)=THR(I)
        PSI(J)=PSR(I)
        PTO(J)=PTR(I)
        IF(I.EQ.1) GO TO 23
        NPTS=3.141593/ARSIN(0.5/FLOAT(I-1))+0.5
        DT=360.0/NPTS
        NPTS=NPTS-1
        IF(NBI.EQ.1) NB(IB)=J
        DO 22 K=1,NPTS
          J=J+1
          X(J)=XR(I)
          Y(J)=YR(I)+FLOAT(K)*DT
          P(J)=PR(I)
          Q(J)=QR(I)
          THETA(J)=THR(I)
          PSI(J)=PSR(I)
          PTO(J)=PTR(I)
          IF(NBI.NE.1) GO TO 22
          IB=IB+1
          IF(IB.GT.200) GO TO 99
          NB(IB)=J
22  CONTINUE
23  CONTINUE
      N=J
24  CONTINUE

```

```

DO 25 I=1,N
ST=SIN(0.01745329*Y(I))
CT=COS(0.01745329*Y(I))
Y(I)=X(I)*ST
X(I)=X(I)*CT
TANP=SQRT((TAN(THETA(I))**2+TAN(PSI(I))**2)
PSI(I)=ATAN(TANP*ST)
THETA(I)=ATAN(TANP*CT*COS(PSI(I)))
25 CONTINUE
26 CONTINUE
ATOT=SQRT(G*RO*TTC)
GM1=G-1.0
GM1G=GM1/G
GP1=G+1.0
GM1H=0.5*GM1
GPGM=GP1/GM1
GGM1=G/GM1
XNB=0.0
YNB=0.0
DO 261 I=1,IB
J=NB(I)
XNB=XNB+X(J)
YNB=YNB+Y(J)
261 CONTINUE
XNB=XNB/IB
YNB=YNB/IB
DO 27 I=1,IB
J=NB(I)
RHO(I)=ATAN2(Y(J)-YNB,X(J)-XNB)
IF(RHO(I).LT.0.0) RHO(I)=6.283185+RHO(I)
NBO(I)=J
27 CONTINUE
CALL SORT(RHO,NBO,IB)
NB(IB+1)=0
NBO(IB+1)=0
DO 30 I=1,N
IF(IMV.EQ.1) GO TO 28
V=Q(I)
IF(IMV.EQ.2) V=SQRT(V**2+THETA(I)**2+PSI(I)**2)
SM=(V/ATOT)**2
SM=SQRT(SM/(1.0-GM1H*SM))
IF(IMV.EQ.0) GO TO 29
PSI(I)=ATAN2(PSI(I),Q(I))
THETA(I)=ATAN2(THETA(I),Q(I)/COS(PSI(I)))
Q(I)=V
GO TO 29
28 CONTINUE
SM=Q(I)
Q(I)=SM*ATOT/SQRT(1.0+GM1H*SM**2)
29 CONTINUE

```

```
SMS=SM**2
SMFAC=1.0+GM1H*SMS
T=TTO/SMFAC
RHO(I)=P(I)/(RO*T)
30 CONTINUE
  IF(IDBUG( 4).NE.0)
    *WRITE(6,6)HEAD,(X(I),Y(I),P(I),RHO(I),Q(I),THETA(I),
    *PSI(I),PTO(I),I=1,N)
    IF((ITYPE.EQ.0).AND.(INEIGH.EQ.0)) INEIGH=1
    IF(INEIGH.EQ.0) CALL NEIGH
    IF(INEIGH.EQ.1) CALL RNEIG
    IF(INEIGH.EQ.2) CALL READNE
    RETURN
99 CONTINUE
  WRITE(6,7)
  STOP
END
```

```

SUBROUTINE NEIGH
COMMON/VALUE/ZE,X(1000),Y(1000),P(1000),RHO(1000) ,
*Q(1000),THETA(1000),PSI(1000),N
INTEGER*2 NE(8,1000)
COMMON /NEAR/NE,DMIN
COMMON/IDBUG/IDBUG(80)
DIMENSION LIST(19),LONG(19)
DATA LIST/1,2,8,20,39,64,95,133,177,227,284,347,416,
*491,573,661,755,856,963/
DATA LONG/0.5,11,18,24,30,37,43,49,56,62,68,74,81,87,
*93,100,106,0/
1 FORMAT('1',20X,'NEIGHBORS',5X,'DMIN =',1PE12.4
*/6X,'BASE',9X,'1',9X,'2',9X,'3',9X,'4',
*9X,'5',9X,'6',9X,'7',9X,'8'//)
2 FORMAT(9I10)
DIMENSION DXM(8)
DMIN=1.E10
DO 23 I=1,N
IC=0
DO 22 J= 1,N
IF(I.EQ.J) GO TO 22
D=(X(I)-X(J))**2+(Y(I)-Y(J))**2
IF(IC.EQ.8) GO TO 21
IC=IC+1
NE(IC,I)=J
DXM(IC)=D
IF(IC.EQ.8) CALL SORT(DXM,NE(1,I),8)
GO TO 22
21 CONTINUE
IF(D.GE.DXM(8)) GO TO 22
NE(8,I)=J
DXM(8)=D
CALL SORT(DXM,NE(1,I),8)
22 CONTINUE
IF(DXM(1).LT.DMIN) DMIN=DXM(1)
23 CONTINUE
GO TO 29
ENTRY RNEIG
J=1
DO 28 I=1,N
IF(1.GE.LIST(J+1)) J=J+1
IF(J.NE.1) GO TO 25
DO 24 K=1,6
NE(K,I)=K+1
24 CONTINUE
NE(7,I)=11
NE(8,I)=17
GO TO 28
25 CONTINUE
LL=I-1

```

```

      IF(LL.LT.LIST(J)) LL=LIST(J+1)-1
      LH=LH+1
      IF(LH.GE.LIST(J+1)) LH=LIST(J)
      NE(1,I)=LL
      NE(2,I)=LH
      IF(J.NE.2) GO TO 26
      LL=LL-1
      IF(LL.LT.LIST(J)) LL=LIST(J+1)-1
      LH=LH+1
      IF(LH.GE.LIST(J+1)) LH=LIST(J)
      NE(3,I)=LL
      NE(4,I)=1
      NE(5,I)=LH
      K=J+1
      GO TO 27
26 CONTINUE
      CALL FIND(I,LIST(J-1),LONG(J-1),NE(3,I))
      K=J+1
      IF(N.LT.LIST(K)) K=J-2
27 CONTINUE
      CALL FIND(I,LIST(K),LONG(K),NE(6,I))
28 CONTINUE
      GO TO 29
      ENTRY READNE
      READ(8,2)(K, (NE(J,K), J=1,8), I=1,N)
29 CONTINUE
      IF( (DEBUG( 5).NE.0)
      *WRITE(6,1)DMIN
      IF( (DEBUG( 5).NE.0)
      *WRITE(6,2)(I, (NE(J,I), J=1,8), I=1,N)
      RETURN
      END

```

```

SUBROUTINE NEWRAP(T,P,A,B,C,N1,N2,N3)
IMPLICIT REAL*8(A-H,O-Z)
REAL*8 N1,N2,N3,J
1 FORMAT('0 FAILED TO CONVERGE IN NEWRAP  T=',1PE12.5,
*5X,'TN=',E12.5,6X,'P=',E12.5,5X,'PN=',E12.5/5X,'TO=',
*E12.5,5X,'PO=',E12.5/6X,'A=',E12.5,6X,'B=',E12.5,6X,
*'C=',E12.5,5X,'N1=',E12.5,5X,'N2=',E12.5,5X,'N3=',
*E12.5)
TO=T
PO=P
DO 22 I=1,50
SINT=DSIN(T)
COST=DCOS(T)
SINP=DSIN(P)
COSP=DCOS(P)
F=A*T+B*P-C
G=N1*COST*COSP+N2*SINT+N3*COST*SINP
DGT=-N1*SINT*COSP+N2*COST-N3*SINT*SINP
DGP=-N1*COST*SINP+N3*COST*COSP
J=A*DGP-B*DGT
TN=T-(F*DGP-G*B)/J
PN=P+(F*DGT-G*A)/J
IF(DABS((T-TN)).GT.1.D-6) GO TO 21
IF(DABS((P-PN)).LE.1.D-6) GO TO 23
21 CONTINUE
IF(I.EQ.50) GO TO 22
T=0.5D0*(T+TN)
P=0.5D0*(P+PN)
22 CONTINUE
WRITE(6,1)T,TN,P,PN,TO,PO,A,B,C,N1,N2,N3
CALL ERRWCA
STOP
23 CONTINUE
T=TN
P=PN
RETURN
END

```

```

SUBROUTINE NORMAL(X,Y,Z,A,B,C,N1,N2,N3,IT,IB)
IMPLICIT REAL*8(A-H,O-Z)
REAL*4 ZC,XC,YC
COMMON /GEOM/ZC(50),XC(50,50),YC(50,50),NS,NP
REAL*8 N1,N2,N3
COMMON/IDEBUG/IDBUG(80)
REAL*8 F(3,5),FA(5),FB(5),FC(5)
1  FORMAT('0 NORMAL X=',1PE12.5,5X,'Y=',1PE12.5,5X,'Z=',
  *E12.5,5X,2HA=',E12.5,5X,'B=',1PE12.5,5X,'C=',1PE12.5)
2  FORMAT(8X,'T1',12X,'T2',12X,'A2',12X,'B2',12X,'C2',
  *12X,'N1',12X,'N2',12X,'N3'/1PE14.5)
  IF(IDBUG(11).NE.0)
    *WRITE(6,1)X,Y,Z,A,B,C
    NSM=NS-1
    DO 21 I=2,NSM
      IF(Z.LE.ZC(I)) GO TO 22
21  CONTINUE
      I=NSM
22  CONTINUE
      CALL BDYFIT(A1,B1,C1,XC(1,I-1),YC(1,I-1),IB,IT1)
      CALL BDYFIT(A2,B2,C2,XC(1,I),YC(1,I),IB,IT2)
      CALL BDYFIT(A3,B3,C3,XC(1,I+1),YC(1,I+1),IB,IT3)
      IF(IT2.EQ.4) GO TO 23
      IF((IT1.LE.IT2).AND.(IT3.LE.IT2)) GO TO 24
      GO TO 27
23  CONTINUE
      IF((IT1.NE.IT2).OR.(IT3.NE.IT2)) GO TO 27
24  CONTINUE
      FAC1=1.D0/(ZC(I)-ZC(I-1))
      FAC2=1.D0/(ZC(I+1)-ZC(I-1))
      FAC3=1.D0/(ZC(I+1)-ZC(I))
      FAC4=ZC(I-1)+ZC(I)
      A2=(A2-A1)*FAC1
      B2=(B2-B1)*FAC1
      C2=(C2-C1)*FAC1
      A3=(A3-A1)*FAC2
      B3=(B3-B1)*FAC2
      C3=(C3-C1)*FAC2
      A3=(A3-A2)*FAC3
      B3=(B3-B2)*FAC3
      C3=(C3-C2)*FAC3
      A2=A2-A3*FAC4
      B2=B2-B3*FAC4
      C2=C2-C3*FAC4
      A1=A1-(A2+A3*ZC(I-1))*ZC(I-1)
      B1=B1-(B2+B3*ZC(I-1))*ZC(I-1)
      C1=C1-(C2+C3*ZC(I-1))*ZC(I-1)
      AX=A1+(A2+A3*Z)*Z
      BY=B1+(B2+B3*Z)*Z
      IF(IT2.NE.4) GO TO 25

```

```

N2=-2.00*(X-A1-(A2+A3*Z)*Z)
N2=-DABS(N2)*DSIGN(1.00,X-AX)
N3=-2.00*(Y-B1-(B2+B3*Z)*Z)
N3=-DABS(N3)*DSIGN(1.00,Y-BY)
N1=-N2*(A2+2.00*A3*Z)- N3*(B2+2.00*B3*Z)+C2+2.00*C3*Z
GO TO 26
25 CONTINUE
N2=-DABS(AX)*DSIGN(1.00,X-AX)
N3=-DABS(BY)*DSIGN(1.00,Y-BY)
N1=-(A2+2.00*A3*Z)*X-(B2+2.00*B3*Z)*Y+(C2+2.00*C3*Z)
26 CONTINUE
D=DSQRT(N1**2+N2**2+N3**2)
N1=N1/D
N2=N2/D
N3=N3/D
IF(10BUG(11).NE.0)
*WRITE(6,2)T1,T2,A2,B2,C2,N1,N2,N3
RETURN
27 CONTINUE
IF(1T1.EQ.4) GO TO 29
F(1,1)=0.00
F(1,2)=A1
F(1,3)=0.00
F(1,4)=B1
F(1,5)=C1
GO TO 30
29 CONTINUE
F(1,1)=1.00
F(1,2)=-2.00*A1
F(1,3)=1.00
F(1,4)=-2.00*B1
F(1,5)=C1-A1**2-B1**2
30 CONTINUE
IF(1T2.EQ.4) GO TO 31
F(2,1)=0.00
F(2,2)=A2
F(2,3)=0.00
F(2,4)=B2
F(2,5)=C2
GO TO 32
31 CONTINUE
F(2,1)=1.00
F(2,2)=-2.00*A2
F(2,3)=1.00
F(2,4)=-2.00*B2
F(2,5)=C2-A2**2-B2**2
32 CONTINUE
IF(1T3.EQ.4) GO TO 33
F(3,1)=0.00
F(3,2)=A3

```

```

      F(3,3)=0.00
      F(3,4)=B3
      F(3,5)=C3
      GO TO 34
33  CONTINUE
      F(3,1)=1.00
      F(3,2)=-2.00*A3
      F(3,3)=1.00
      F(3,4)=-2.00*B3
      F(3,5)=C3-A3**2-B3**2
34  CONTINUE
      DZ21=Z2-Z1
      DZ31=Z3-Z1
      DZ32=Z3-Z2
      Z12=Z1+Z2
      DO 35 K=1,5
      FB(K)=(F(2,K)-F(1,K))/DZ21
      FC(K)=(F(3,K)-F(1,K))/DZ31-FB(K)/DZ32
      FB(K)=FB(K)-FC(K)*Z12
      FA(K)=F(1,K)-(FB(K)+FC(K)*Z1)*Z1
35  CONTINUE
      N2=2.00*(FA(1)+(FB(1)+FC(1)*Z)*Z)*X+FA(2)+(FB(2)+FC(2)*Z)*Z
      N3=2.00*(FA(3)+(FB(3)+FC(3)*Z)*Z)*Y+FA(4)+(FB(4)+FC(4)*Z)*Z
      N1=-(FB(1)+2.00*FC(1)*Z)*X**2-(FB(2)+2.00*FC(2)*Z)*X
      *   -(FB(3)+2.00*FC(3)*Z)*Y**2-(FB(4)+2.00*FC(4)*Z)*Y
      *   +(FB(5)+2.00*FC(5)*Z)
      AX=FA(1)+(FB(1)+FC(1)*Z)*Z
      IF(AX.EQ.0.00) AX=-2.00
      AX=-2.00*(FA(2)+(FB(2)+FC(2)*Z)*Z)/AX
      BY=FA(3)+(FB(3)+FC(3)*Z)*Z
      IF(BY.EQ.0.00) BY=-2.00
      BY=-2.00*(FA(4)+(FB(4)+FC(4)*Z)*Z)/BY
      N2=-DABS(N2)*DSIGN(1.00,X-AX)
      N3=-DABS(N3)*DSIGN(1.00,Y-BY)
      GO TO 26
      END

```

```

SUBROUTINE OUTPUT(IT)
COMMON /TITLE/HEAD(20)
COMMON/VALUE/ZE,X(1000),Y(1000),P(1000),RHO(1000) ,
*Q(1000),THETA(1000),PSI(1000),N
COMMON /STAG/PTO(1000),TTO,ATOT,ZO,ZMAX,NB(200),
*NBO(200),JMAX,JPT,IB,IPLLOT,IT11,IT12
COMMON /GAS/G,RO,GM1,GM1H,GM1G,GGM1,GP1,GPGM
INTEGER*2 NB,NBC
1 FORMAT('1',20X,20A4//30X,'ZE =' ,1PE12.5,15X,'STEP NO.'
*,15//2X,'NO.',8X,'X',14X,'Y',14X,'P',13X,'RHO',13X,'Q'
*,12X,'THETA',11X,'PSI',13X,'M',8X,'BODY'//)
2 FORMAT(1X,I4,1P8E15.5,1X,A4)
3 FORMAT('1',20X,20A4//10X,'BODY POINTS AT ZE =' ,1PE12.5
*,15X,'STEP NO.',15//2X,'NO.',8X,'X',14X,'Y',14X,'P',
*,13X,'RHO',13X,'Q',12X,'THETA',11X,'PSI',13X,'M'//)
4 FORMAT(1X,I4,1P8E15.5)
DATA NES/' YES'//,NO/' '/
DIMENSION XP(1002),YP(1002)
DATA XO/0.0/,DX/25.0/,YO/10.0/,DY/5.0/
IF(IPLLOT.EQ.0) GO TO 21
F=0.0
DO 20 I=1,N
XP(I)=F
YP(I)=P(I)
F=F+1.0
20 CONTINUE
XP(N+1)=XO
XP(N+2)=DX
YP(N+1)=YO
YP(N+2)=DY
CALL AXIS(0.0,0.0,0.0,'POINT',-5,14.0,0.0,XO,DX,10.0)
CALL AXIS(0.0,0.0,0.0,'P',1,10.0,90.0,YO,DY,10.0)
CALL LINE(XP,YP,N,1.0,1)
CALL SYMBOL(8.0,8.0,0.15,'STEP ',0.0,0.5)
CALL NUMBER(999.0,999.0,0.15,FLOAT(IT),0.0,-1)
CALL CALCMP(0.0,0.0,0.0,2)
21 CONTINUE
IF(IT11.EQ.0) GO TO 22
WRITE(11)ZE,N,IB,NB,NBO,(X(J),Y(J),P(J),RHO(J),Q(J),
*THETA(J),PSI(J),PTO(J),J=1,N)
22 CONTINUE
IF(IT12.EQ.0) GO TO 24
WRITE(12)ZE,N,IB,NB,NBO
DO 23 J=1,IB
K=NBO(J)
WRITE(12)X(K),Y(K),P(K),RHO(K),Q(K),THETA(K),PSI(K)
23 CONTINUE
24 CONTINUE
IF(JPT.EQ.1) GO TO 25
IF((IT.EQ.JMAX).OR.(ZE.EQ.ZMAX)) GO TO 25

```

```

      IF(MOD(IT,JPT).NE.0) RETURN
25 CONTINUE
      L=1
      DO 28 J=1,N,50
      N2=J+49
      IF(N2.GT.N) N2=N
      WRITE(6,1)HEAD,ZE,IT
      DO 27 K=J,N2
      SM=Q(K)/SQRT(G*P(K)/RHO(K))
      IN=NO
      IF(NB(L).NE.K) GO TO 26
      IN=NE S
      L=L+1
26 CONTINUE
      WRITE(6,2)K,X(K),Y(K),P(K),RHO(K),Q(K),THETA(K),
      *PSI(K),SM,IN
27 CONTINUE
28 CONTINUE
      WRITE(6,3)HEAD,ZE,IT
      DO 29 J=1,18
      K=NB0(J)
      SM=Q(K)/SQRT(G*P(K)/RHO(K))
      WRITE(6,4)K,X(K),Y(K),P(K),RHO(K),Q(K),THETA(K),
      *PSI(K),SM
29 CONTINUE
      RETURN
      END

```

```

SUBROUTINE SIMQ(A,R,M,N)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION A(1),R(1)
DATA EPS/1.D-5/
111 FORMAT('0 SINGULAR MATRIX IN SIMQ')
IF(M)23,23,1
C SEARCH FOR GREATEST ELEMENT IN MATRIX A
1 IER=0
PIV=0.00
MM=M*M
NM=N*M
DO 3 L=1,MM
TB=DABS(A(L))
IF(TB-PIV)3,3,2
2 PIV=TB
I=L
3 CONTINUE
TOL=EPS*PIV
C A(I) IS PIVOT ELEMENT. PIV CONTAINS /A(I)/.
C START ELIMINATION LOOP
LST=1
DO 17 K=1,M
C TEST ON SINGULARITY
IF(PIV)23,23,4
4 IF(IER)7,5,7
5 IF(PIV-TOL)6,6,7
6 IER=K-1
7 PIV=1.00/A(I)
J=(I-1)/M
I=I-J*M-K
J=J+1-K
C I+K IS ROW-INDEX, J+K COLUMN-INDEX OF PIVOT ELEMENT
C PIVOT ROW REDUCTION AND ROW INTERCHANGE IN R.
DO 8 L=K,NM,M
LL=L+I
TB=PIV*R(LL)
R(LL)=R(L)
8 R(L)=TB
C IS ELIMINATION TERMINATED
IF(K-M)9,18,18
C COLUMN INTERCHANGE IN MATRIX A
9 LEND=LST+M-K
IF(J)12,12,10
10 II=J*M
DO 11 L=LST,LEND
TB=A(L)
LL=L+II
A(L)=A(LL)
11 A(LL)=TB
C ROW INTERCHANGE AND PIVOT ROW REDUCTION IN MATRIX A

```

```

12 DO 13 L=LST,MM,M
    LL=L+1
    TB=PIV*A(LL)
    A(LL)=A(L)
13 A(L)=TB
C    SAVE COLUMN INTERCHANGE INFORMATION
    A(LST)=J
C    ELEMENT REDUCTION AND NEXT PIVOT SEARCH
    PIV=0.00
    LST=LST+1
    J=0
    DO 16 II=LST,LEND
        PIV=-A(II)
        IST=II+M
        J=J+1
    DO 15 L=IST,MM,M
        LL=L-J
        A(L)=A(L)+PIV*A(LL)
        TB=DABS(A(L))
        IF(TB-PIV)15,15,14
14 PIV=TB
    I=L
15 CONTINUE
    DO 16 L=K,NM,M
        LL=L+J
16 R(LL)=R(LL)+PIV*R(L)
17 LST=LST+M
C    END OF ELIMINATION LOOP
C    BACK SUBSTITUTION AND BACK INTERCHANGE
18 IF(M-1)23,22,19
19 IST=MM+M
    LST=M+1
    DO 21 I=2,M
        II=LST-I
        IST=IST-LST
        L=IST-M
        L=A(L)+0.5D0
        DO 21 J=II,NM,M
            TB=R(J)
            LL=J
            DO 20 K=IST,MM,M
                LL=LL+1
20 TB=TB-A(K)*R(LL)
                K=J+L
                R(J)=R(K)
21 R(K)=TB
22 RETURN
C    ERROR RETURN
23 IER=-1
    WRITE(6,111)
    STOP
    END

```

```

SUBROUTINE SOLVBP(X,Y,A,B,C,A1,B1,C1,IT,X1,Y1)
  IMPLICIT REAL*8(A-H,O-Z)
  1 FORMAT('0 NEGATIVE RADICAL IN SOLVBP')
  GO TO(21,22,23,24),IT
21 CONTINUE
  X=C1
  Y=(C-A*X)/B
  RETURN
22 CONTINUE
  Y=C1
  X=(C-B*Y)/A
  RETURN
23 CONTINUE
  Y=(A*C1-C*A1)/(A*B1-B*A1)
  X=(C-B*Y)/A
  RETURN
24 CONTINUE
  IF(DABS(B).GT.DABS(A)) GO TO 26
  D1=1.D0+(B/A)**2
  D2=C/A-A1
  D3=(D2**2+B1**2-C1)/D1
  D2=(B*D2/A+B1)/D1
  RAD=D2**2-D3
  IF(RAD.GE.0.D0) GO TO 25
  WRITE(6,1)
  WRITE(6,2)X,Y,A,B,C,A1,B1,C1,IT,X1,Y1,D1,D2,D3
2  FORMAT('0  X=',1PE12.5,5X,'Y=',E12.5/3X,'A=',E12.5,5X,
  *'B=',E12.5,5X,'C=',E12.5/2X,'A1=',E12.5,4X,'B1=',E12.5,
  *'4X,'C1=',E12.5/2X,'IT=',I2,14X,'X1=',E12.5,4X,'Y1=',
  *E12.5/2X,'D1=',E12.5,4X,'D2=',E12.5,4X,'D3=',E12.5)
  CALL ERRWCA
  STOP
25 CONTINUE
  RAD=DSQRT(RAD)
  Y=D2+RAD
  IF(DABS(Y1-Y).GE.DABS(Y1-D2+RAD)) Y=D2-RAD
  X=(C-B*Y)/A
  RETURN
26 CONTINUE
  D1=1.D0+(A/B)**2
  D2=C/B-B1
  D3=(D2**2+A1**2-C1)/D1
  D2=(A*D2/B+A1)/D1
  RAD=D2**2-D3
  IF(RAD.GE.0.D0) GO TO 27
  WRITE(6,1)
  WRITE(6,2)X,Y,A,B,C,A1,B1,C1,IT,X1,Y1,D1,D2,D3
  CALL ERRWCA
  STOP
27 CONTINUE

```

```
RAD=DSQRT(RAD)
X=D2+RAD
IF(DABS(X1-X),GT.DABS(X1-D2+RAD)) X=D2-RAD
Y=(C-A*X)/B
RETURN
END
```

```
SUBROUTINE SORT(DN,NE,N)
DIMENSION DN(N)
INTEGER*2 NE(N)
N1=1
N2=N
20 CONTINUE
  DMIN=DN(N1)
  DMAX=DN(N2)
  IF(DMIN.LE.DMAX) GO TO 21
  D=DN(N1)
  DN(N1)=DN(N2)
  DN(N2)=D
  M=NE(N1)
  NE(N1)=NE(N2)
  NE(N2)=M
  GO TO 20
21 CONTINUE
  N1=N1+1
  N2=N2-1
  IF(N1.GT.N2) GO TO 24
  DO 23 I=N1,N2
    IF(DN(I).GE.DMIN) GO TO 22
    DN(N1-1)=DN(I)
    DN(I)=DMIN
    DMIN=DN(N1-1)
    M=NE(I)
    NE(I)=NE(N1-1)
    NE(N1-1)=M
    GO TO 23
22 CONTINUE
  IF(DN(I).LE.DMAX) GO TO 23
  DN(N2+1)=DN(I)
  DN(I)=DMAX
  DMAX=DN(N2+1)
  M=NE(I)
  NE(I)=NE(N2+1)
  NE(N2+1)=M
23 CONTINUE
  GO TO 20
24 CONTINUE
  RETURN
END
```

```

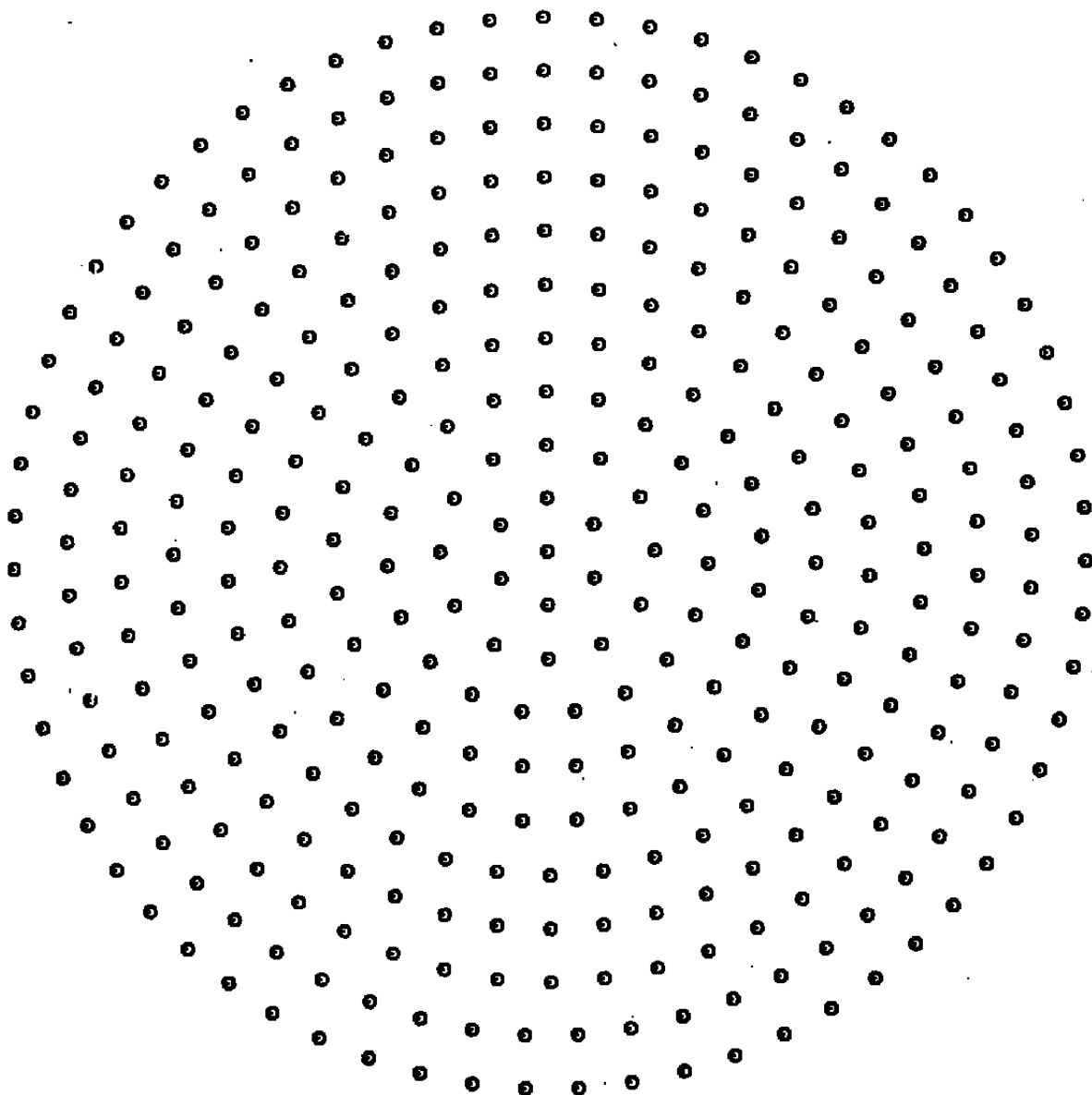
SUBROUTINE THREEED
COMMON/VALUE/ZE,X(1000),Y(1000),P(1000),RHO(1000) ,
*Q(1000),THETA(1000),PSI(1000),N
COMMON /GAS/G,RQ,GM1,GM1H,GM1G,GGM1,GP1,GPGM
COMMON/NVALU/DZ,XN(1000),YN(1000),PN(1000),RN(1000),
*QN(1000),TN(1000),PSN(1000)
COMMON /BCUT/XB(50),YB(50)
COMMON /OCUT/XBO(50),YBO(50)
COMMON /STAG/PTO(1000),TTO,ATOT,ZO,ZMAX,NB(200),
*NBO(200),JMAX,JPT ,IB
INTEGER*2 NB,NBO
REAL*8 ZO,DG,DRC,DGM1,DGM1H,DGM1G,DGGM1,DGP1,DGPGM
COMMON /DGAS/ZO,DG,DRO,DGM1,DGM1H,DGM1G,DGGM1,DGP1,
*DGPGM
REAL*8 PTOD,TTOD,ATOD
COMMON /DSTAG/PTOD(1000),TTOD,ATOD
DO 19 I=1,N
PTOD(I)=PTC(I)
19 CONTINUE
TTOD=TTO
ATOD=ATOT
DG=G
DRO=RO
DGM1=DG-1.00
DGM1H=0.500*DGM1
DGM1G=DGM1/DG
DGGM1=1.00/DGM1G
DGP1=DG+1.00
DGPGM=DGP1/DGM1
CALL CUT(ZO)
ZE=ZO
DO 20 I=1,50
XBO(I)=XB(I)
YBO(I)=YB(I)
20 CONTINUE
DO 26 J=1,JMAX
CALL OUTPUT(J-1)
CALL DIST(DS)
DZ= 0.5*DS
IF((ZE+DZ).LE.ZMAX) GO TO 21
IF(ZE.GE.ZMAX) RETURN
DZ=ZMAX-ZE
21 CONTINUE
ZO=DZ
CALL CUT(ZE+DZ)
K=1
DO 23 I=1,N
CALL FIT(I)
IF(NB(K).EQ.I) GO TO 22
CALL FIELD(I)

```

```
      GO TO 23
22  CONTINUE
      CALL BODY(I)
      K=K+1
23  CONTINUE
      ZE=ZE+DZ
      DO 24 I=1,N
        X(I)=XN(I)
        Y(I)=YN(I)
        P(I)=PN(I)
        RHO(I)=RN(I)
        Q(I)=QN(I)
        THETA(I)=TN(I)
        PSI(I)=PSN(I)
24  CONTINUE
      DO 25 I=1,50
        XB0(I)=XB(I)
        YB0(I)=YB(I)
25  CONTINUE
26  CONTINUE
      CALL OUTPUT(JMAX)
      RETURN
      END
```

APPENDIX B EXAMPLE PROBELM

THE SAMPLE PROBLEM IS FOR $M = 4.1$ AXISYMMETRIC NOZZLE WITH $\gamma = 1.2$. THE NOZZLE AXIS IS ALIGNED WITH THE z AXIS. THERE ARE 48 STATIONS FOR THE BODY GEOMETRY. THE STARTING PLANE INPUT HAS ONE RAY WITH 11 POINTS. THE COMPLETE STARTING PLANE IS GENERATED FROM THESE 11 POINTS TO A TOTAL OF 346 NEARLY EQUALLY SPACED POINTS. THE ARRANGEMENT OF THESE POINTS IS SHOWN BELOW.



INPUT CARDS

```
0000000001111111112222222223333333334444444445555555556666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890
```

MACH & NOZZLE (G=1.24)

```
0
45
36 3 0.205
1.01532 0.0
36 3 0.25
1.02299 0.0
36 3 0.30
1.03344 0.0
36 3 0.35
1.04594 0.0
36 3 0.40
1.06051 0.0
36 3 0.45
1.07710 0.0
36 3 0.50
1.09561 0.0
36 3 0.55
1.11586 0.0
36 3 0.60
1.13762 0.0
36 3 0.65
1.16058 0.0
36 3 0.70
1.18442 0.0
36 3 0.80
1.23355 0.0
36 3 0.90
1.28315 0.0
36 3 1.00
1.33273 0.0
36 3 1.10
1.38226 0.0
36 3 1.20
1.43171 0.0
36 3 1.30
1.48105 0.0
36 3 1.40
1.53028 0.0
36 3 1.50
1.57936 0.0
36 3 1.60
1.62831 0.0
36 3 1.70
1.67711 0.0
36 3 1.80
1.72574 0.0
36 3 1.93260
1.78990 0.0
36 3 2.09378
```

```
0000000001111111112222222223333333334444444445555555556666666667777777778
1234567890123456789012345678901234567890123456789012345678901234567890
```

INPUT CARDS

```
00000000011111111122222222233333333344444444455555555566666666677777777778
1234567890123456789012345678901234567890123456789012345678901234567890
```

```
1.86711 0.0
36 3 2.28244
1.95576 0.0
36 3 2.50060
2.05562 0.0
36 3 2.75167
2.16697 0.0
36 3 3.04016
2.29026 0.0
36 3 3.37131
2.42576 0.0
36 3 3.75138
2.57362 0.0
36 3 4.16794
2.73284 0.0
36 3 4.68976
2.90607 0.0
36 3 5.28701
3.08965 0.0
36 3 5.93207
3.28351 0.0
36 3 6.69778
3.48569 0.0
36 3 7.57930
3.69348 0.0
36 3 8.59135
3.92291 0.0
36 3 9.74737
4.10866 0.0
36 3 11.0567
4.30501 0.0
36 3 12.5192
4.48414 0.0
36 3 14.1196
4.63912 0.0
36 3 15.8202
4.76413 0.0
36 3 17.5562
4.85610 0.0
36 3 19.2369
4.91634 0.0
36 3 20.7600
4.95043 0.0
36 3 22.0350
4.98647 0.0
36 3 23.0095
4.97238 0.0
36 3 24.0
4.97416 0.0
1.24 100.0 100.0 1710.0
```

```
00000000011111111122222222233333333344444444455555555566666666677777777778
1234567890123456789012345678901234567890123456789012345678901234567890
```

INPUT CARDS

000000000111111112222222222333333333444444444555555555666666666777777777
12345678901234567890123456789012345678901234567890123456789012345678901234567890

0.205	24.0	0	1	11	36	100	100				
0.0	0.0	55.035	1.0105	0.0	0.0	0.0	0.0	0			
0.10153	0.0	54.887	1.01291	0.22466	0.0	0.0	0.0	0			
0.20306	0.0	54.439	1.02022	0.47564	0.0	0.0	0.0	0			
0.30460	0.0	53.676	1.03271	0.77981	0.0	0.0	0.0	0			
0.40613	0.0	52.571	1.05091	1.16897	0.0	0.0	0.0	0			
0.50766	0.0	51.079	1.07571	1.68158	0.0	0.0	0.0	0			
0.60919	0.0	49.127	1.10859	2.36939	0.0	0.0	0.0	0			
0.71072	0.0	46.613	1.15183	3.30375	0.0	0.0	0.0	0			
0.81226	0.0	43.385	1.20905	4.58134	0.0	0.0	0.0	0			
0.91379	0.0	39.245	1.28600	6.32552	0.0	0.0	0.0	0			
1.01532	0.0	33.953	1.39283	8.67814	0.0	0.0	0.0	1			

000000000111111112222222222333333333444444444555555555666666666777777777
12345678901234567890123456789012345678901234567890123456789012345678901234567890

MACH & NOZZLE (G=1.24)

STARTING PLANE INPUT

GAMMA = 1.240	PTD = 1.00000E 02	TTO = 1.00000E 02	R = 1.71600E 03			
ZD= 2.05000E-01	ZMAX= 2.40000E 01	ITYPE= 0	IMV= 1	NPTS= 11	NRAYS= 36	JMAX= 100
		IPLOT= 0	IT11= 0	IT12= 0	INEIGH= 0	JPT= 100
X	Y	P	Q	THETA	PSI	PTD

337	5.50853E-01	-8.52898E-01	3.39530E 01	2.43870E-04	5.78388E 02	8.19522E-02	-1.27519E-01	1.39203E 00
338	6.33038E-01	-7.93812E-01	3.39530E 01	2.43870E-04	5.78388E 02	9.42131E-02	-1.18770E-01	1.39203E 00
339	7.08931E-01	-7.26836E-01	3.39530E 01	2.43870E-04	5.78388E 02	1.05548E-01	-1.08832E-01	1.39203E 00
340	7.77776E-01	-6.52640E-01	3.39530E 01	2.43870E-04	5.78388E 02	1.15842E-01	-9.77970E-02	1.39203E 00
341	8.38893E-01	-5.71955E-01	3.39530E 01	2.43870E-04	5.78388E 02	1.24990E-01	-8.57698E-02	1.39203E 00
342	8.91674E-01	-4.85585E-01	3.39530E 01	2.43870E-04	5.78388E 02	1.32900E-01	-7.28676E-02	1.39203E 00
343	9.35590E-01	-3.94392E-01	3.39530E 01	2.43870E-04	5.78388E 02	1.39487E-01	-5.92188E-02	1.39203E 00
344	9.70210E-01	-2.99276E-01	3.39530E 01	2.43870E-04	5.78388E 02	1.44684E-01	-4.49593E-02	1.39203E 00
345	9.95188E-01	-2.01186E-01	3.39530E 01	2.43870E-04	5.78388E 02	1.48436E-01	-3.02347E-02	1.39203E 00
346	1.01027E 00	-1.01097E-01	3.39530E 01	2.43870E-04	5.78388E 02	1.50703E-01	-1.51965E-02	1.39203E 00

337	2.69695E 00	-4.18475E 00	3.04682E-01	5.44889E-06	1.09336E 03	-8.23262E-04	-1.14736E-04	4.15237E 00
338	3.08550E 00	-3.89932E 00	3.04407E-01	5.44515E-06	1.09340E 03	-1.05458E-03	-3.91975E-04	4.15265E 00
339	3.45482E 00	-3.57631E 00	3.04355E-01	5.44440E-06	1.09341E 03	-1.19741E-03	-8.69508E-04	4.15296E 00
340	3.79389E 00	-3.21509E 00	3.03941E-01	5.43843E-06	1.09346E 03	-1.24832E-03	-9.29814E-04	4.15376E 00
341	4.10187E 00	-2.81117E 00	3.03909E-01	5.43797E-06	1.09348E 03	-7.68471E-04	-5.02632E-04	4.15383E 00
342	4.37311E 00	-2.36708E 00	3.03921E-01	5.43814E-06	1.09348E 03	1.17415E-04	9.81998E-04	4.15380E 00
343	4.59270E 00	-1.92070E 00	3.02732E-01	5.42099E-06	1.09368E 03	-7.02881E-05	7.26741E-04	4.15614E 00
344	4.75201E 00	-1.46527E 00	3.03329E-01	5.42980E-06	1.09358E 03	-2.49965E-04	3.29101E-04	4.15496E 00
345	4.87438E 00	-9.83505E-01	3.03316E-01	5.42942E-06	1.09358E 03	-3.17399E-04	2.13969E-04	4.15499E 00
346	4.94792E 00	-4.94316E-01	3.03261E-01	5.42861E-06	1.09359E 03	-3.46455E-04	9.48671E-05	4.15510E 00

AEDC-TR-78-88

NOMENCLATURE

a	Speed of sound
h	Static enthalpy
L	Distance along bicharacteristic
$\bar{L}, \bar{M}, \bar{N}$	Bicharacteristic coordinate system
M	Mach number
n_1, n_2, n_3	Unit normal to body surface in z, x, y directions
p	Pressure
\bar{q}	Velocity
q	$ \bar{q} $
r^*	Throat radius of axisymmetric nozzle
R	Gas constant
T	Temperature
u	x component of velocity
v	y component of velocity
w	z component of velocity
x, y, z	Cartesian coordinate system

β	Mach angle
γ	Ratio of specific heats
δ	Parametric angle for bicharacteristic
θ, ψ	Flow angles defined in Fig. 1
ρ	Density

SUBSCRIPTS

i	$i = 1, 2, \dots$ refer to quantities at given points
ts	Stagnation conditions along streamline